z/OS and z/VM
2.5

Hardware Configuration Manager User's Guide



SC34-2670-50

Note

Before using this information and the product it supports, read the information in “Notices” on page 441.

This edition applies to Version 2 Release 5 of IBM® z/OS® (5650-ZOS), Version 7 Release 2 of z/VM (5741-A09) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2021-09-30

© Copyright International Business Machines Corporation 1995, 2021.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures ..... **xiii**

Tables ..... **xxi**

About this document ..... **xxiii**
Who this document is for xxiii
How this document is organized xxiii
Related information xxiv

How to send your comments to IBM ..... **xxvii**
If you have a technical problem xxvii

Summary of changes ..... **xxix**
Summary of changes for z/OS Version 2 Release 5 (V2R5) xxix
Summary of changes for z/OS Version 2 Release 4 (V2R4) xxix
Summary of changes for z/OS Version 2 Release 3 (V2R3) xxix
Changes made in z/VM Version 6 Release 2 xxx
Changes made in z/VM Version 6 Release 1 xxx

Chapter 1. Overview **1**
Physical and logical data in one place 1
Easy to use 1
Accurate system configuration 2
Support for different user groups 2

Chapter 2. Installing HCM **3**
Prerequisites 3
Workstation hardware requirements 3
Software requirements 3
Installing HCM on your workstation 4
Purpose and location of the HCM INI file 5
Installation problems 5
Installing service levels 6
Installing HCM in a LAN environment 6
Establishing the host communication 6
Setting up TCP/IP definitions for z/OS 6
Setting up TCP/IP definitions for z/VM 10
Verifying TCP/IP host communication 19
Encrypting the host communication for z/OS 19
TCP/IP problem determination 23
Uninstalling HCM 23

Chapter 3. Getting started **25**
Starting and stopping HCM for z/OS 25
Starting and stopping HCM for z/VM 26
Working with the user interface 27
Keyboard 27
Mouse 28
Dialogs 28

Online help..... 29
User data fields..... 29
Messages..... 31
Maintaining the configuration diagram..... 33
Filtering your view..... 33
Connections in the diagram..... 35
Saving views..... 36
Vertical view of XMP processors..... 37
Visualizing ESCON or FICON switch and crossbar switch configurations..... 38
Showing page breaks..... 40
Zooming the view..... 41
Scrolling the view..... 41
Controlling text detail..... 41
Setting preferences..... 41
Showing screen elements..... 42
Specifying options..... 42
Receiving visual warnings..... 42
Printing the configuration diagram..... 43
Selecting objects..... 43
Locating objects..... 43
Filtering objects..... 45
Creating objects..... 45
Editing objects..... 45
Deleting objects..... 46
Reversing an action..... 46
Ordering objects..... 46

Chapter 4. Building your configuration..... 49
HCM and HCD..... 49
Relation of configuration files and IODFs..... 50
Data integrity between IODF and configuration file..... 52
The IODF load / resync process..... 52
Introduction to the master configuration file concept..... 54
Physical and logical objects in HCM..... 54
Graphical representation of physical objects..... 55
Creating a new configuration from an IODF..... 56
Preparing the IODF..... 57
Loading the IODF..... 59
Creating a new configuration from scratch..... 66
Opening an existing configuration..... 67
Opening an existing configuration file and associated IODF..... 67
Opening an existing configuration file with a new IODF..... 68
Upgrading configuration files..... 69
Working with your IODFs and configuration files..... 71
Copying an existing IODF..... 71
Making a work or a production IODF..... 72
Using the CHPID Mapping Tool Support utility..... 73
Creating work and production IODFs without the associated configuration files..... 74
Exporting and importing IODFs..... 75
Deleting an existing IODF..... 75
Checking a configuration..... 76
Copying a configuration..... 76
Creating work and production IODFs with the associated configuration files..... 76
Compressing configuration files..... 78
Naming convention of your configuration files and IODFs..... 78
Deleting a configuration file..... 78
Saving and closing your configuration..... 78

Using the master configuration file (MCF)....79
Enabling/disabling a configuration for MCF-sharing....80
Granting access to the MCF on the host....81
How to work with MCF-enabled configurations: Scenarios....82
How to resolve physical mismatches in a shared configuration....86
How to work with configuration packages....88
How to create and edit a configuration package....89
How to work with configuration package objects....90
How to transmit a configuration package....92

Chapter 5. Stand-alone mode....95
Invoking the stand-alone mode....95
Working with the stand-alone mode....95
Creating objects....95
Viewing objects....95
Printing reports, diagrams, cable labels....96
No host connection....96
Work configuration file....96
Production configuration file....96
Upgrading....97
Resynchronizing....97
VOLSER reconciliation....97

Chapter 6. Modifying objects in your configuration....99
General concepts of the copy wizards....99
How HCM handles connections when using the copy wizards....100
Mapping CHPIDs....100
Connections in the Upgrade scenario....101
Connections in the Repeat scenario....104
Processors....106
Creating a processor....108
Copying a processor using a wizard....109
Editing a processor....110
Deleting a processor....110
Managing PCIe functions....111
Channel subsystems....113
Creating or editing a channel subsystem....113
Copying a channel subsystem....113
Deleting a channel subsystem....114
Partitions....114
Creating a partition....115
Copying a partition....116
Editing a partition....116
Deleting a partition....117
CHPIDs....117
CHPID interface type....119
CHPID operation mode....120
CHPID operation mode symbols....120
Unmanaged/managed CHPIDs....121
Selecting a CHPID to view details....121
Create CHPIDs....122
Editing a CHPID....126
Deleting a CHPID....126
Assigning CHPIDs to partitions....126
Support of spanned CHPIDs....128
Aggregating CHPIDs....130
Controllers....130

Creating a controller....131
Editing a controller....132
Editing the fiber links above the controller....137
Changing the channel interface type....137
Deleting a controller....138
Control units....138
Creating a control unit....139
Editing a control unit....139
Deleting a control unit....139
Moving an existing switch 's control unit into a switch....140
Strings....140
Creating a string....141
Editing a string....142
Deleting a string....145
Units....145
Creating a unit....145
Editing a unit....145
Deleting a unit....146
I/O devices....146
Creating an I/O device in a string....147
Editing an I/O device....148
Defining devices to OS configurations....148
Deleting an I/O device....149
Changing the subchannel set for devices....149
Converters....150
Creating a converter....151
Deleting a converter....152
Editing a converter....152
Switches....154
Creating a switch....155
Editing a switch....155
Deleting a switch....158
Managing switch configurations....158
Crossbar switches....162
Creating a crossbar switch....162
Editing a crossbar switch....162
Visualizing a crossbar switch configuration....165
Deleting a crossbar switch....165
Cabinets....165
Creating a cabinet....166
Editing a cabinet....166
Deleting a cabinet....167
Patchports....167
Cabinet dialog controls....168
Syntax of the user-defined patchport/general box port naming format....170
General boxes....171
Creating a general box....171
Editing a general box....172
Deleting a general box....173

Chapter 7. Connecting objects in your configuration....175
Connecting objects....175
Positioning objects along daisy-chained connections....176
Connecting devices to controllers....177
Creating a string under a controller....178
Selecting an existing string under a controller....178
Connecting the string to a controller device interface....179

Disconnecting the string interface from a device interface..... 183
Connecting controllers to processors and switches.....183
Selecting a channel interface for a controller..... 183
Connecting the channel interface to a CHPID.....184
Disconnecting the channel interface from a CHPID.....191
Connecting controllers to controllers (PPRC connections)..... 191
Physically connecting a channel interface to a channel interface of another controller..... 192
Creating a PPRC connection..... 192
Editing a PPRC connection.....192
Importing and exporting PPRC connections..... 192
Deleting a PPRC connection..... 193
Disconnecting a channel interface from a channel interface..... 193
Connecting switches to processors or other switches..... 193
Disconnecting the switch port from a target interface.....195
Connecting CHPIDs via CTC connections..... 195
Creating a CTC connection.....196
Editing a CTC connection..... 197
Validating CTC connections..... 201
Connecting CHPIDs via coupling facility connections..... 203
Creating a coupling facility connection.....205
Deleting a coupling facility connection.....206

Chapter 8. Physical connections.....207
Managing physical cables.....207
Assigning cables to connections.....207
Creating and editing cables..... 209
Printing cable labels.....211
Editing fiber links..... 212
Infrastructure connections..... 212
Disconnecting permanent connections from patchports..... 214
Jumper connections..... 215
Editing fiber links in the dialog.....215
Adding a patchport to the fiber link..... 216
Removing a patchport from the fiber link.....216
Assigning cables between the objects in a fiber link..... 216
Disconnecting a fiber link..... 216
Connecting crossbar switches.....218
Connecting a crossbar switch..... 218
Positioning a crossbar switch on a daisy-chain.....219
Disconnecting a crossbar channel interface..... 219

Chapter 9. Utilities.....221
Performing I/O Autoconfiguration.....221
How to start I/O Autoconfiguration..... 222
How to define the scope and mode of discovery..... 222
How to work with discovered controllers..... 223
How to work with proposed control units..... 224
How to work with proposed devices.....225
How to apply your own numbering scheme for PAV devices.....226
How to work on the Summary page.....227
How to define autoconfiguration policies.....228
How changed control units may affect HCM..... 231
Moving port connections..... 231
Implied ports.....232
Procedure..... 233
Aggregating CHPIDs..... 233
Converting CNC channels to FICON channels..... 233

Changing the CHPID link address format ... 234
Creating/Copying/Editing an I/O subsystem ... 234
Invocation ... 234
Using the utility to create an I/O subsystem ... 234
Using the utility to copy an I/O subsystem ... 235
Using the utility to edit an I/O subsystem ... 235
Exporting IOCP or I/O configuration statements ... 235
Importing I/O configuration statements ... 236
Browsing performance data ... 237
Objects that provide performance data ... 240
Using the WWPN Prediction Tool Support ... 242

Chapter 10. Operating system configurations ... 245
Listing the OS configuration-EDT-esoteric hierarchy ... 245
Creating an OS configuration ... 246
Editing an OS configuration ... 247
Copying an OS configuration ... 247
Deleting an OS configuration ... 247
Listing devices for an OS configuration ... 247
Listing consoles for an OS configuration ... 252
Eligible device tables (EDTs) ... 255
Creating an EDT ... 255
Editing an EDT ... 255
Copying an EDT ... 255
Deleting an EDT ... 255
Esoteric device groups (esoterics) ... 256
Creating an esoteric ... 256
Editing an esoteric ... 256
Copying an esoteric ... 256
Deleting an esoteric ... 256
Listing devices for an esoteric ... 257
Generic device groups (generics) ... 258
Listing system-defined generics ... 259
Listing user-defined generics ... 260

Chapter 11. Reports and diagrams ... 263
Printing configuration diagrams ... 263
Generating text reports and diagram reports ... 264
Sample text reports and diagram reports ... 266
Sample cable labels ... 276
Viewing HCD configuration reports ... 278
Viewing HCD compare reports ... 279
Comparing HCM configurations ... 281

Chapter 12. Scenarios ... 285
Scenario I – Adding and connecting new equipment ... 285
Add four new channels to an existing processor ... 285
Connect the new CHPIDs to patchports ... 288
Add a new FICON switch ... 290
Connect the new switch to one of the new CHPIDs ... 291
Connect another switch to the second new CHPID ... 292
Add a new controller ... 292
Connect the controller to CHPIDs ... 293
Connect the controller to the FICON switches ... 295
Define the fiber links for the connections ... 297
Add some strings to the new controller ... 298
Define the devices to the operating system ... 300

Scenario II – Defining a CTC connection....302
Connect a switch to the CHPIDs....302
Create the first CTC connection....304
Create the second CTC connection....306
Create the third CTC connection....307
Create the last CTC connection....308
Define the CTC devices to the OS configuration....308
Scenario III – Defining a fiber cabling infrastructure....311
Create the cabinets....311
Trunk the two cabinets together....311
Trunk the zone cabinet to the processor....312
Connect a switch to CHPIDs through the patchport cabinets....315
Assign a cable to the new connection....317
Connect a controller to the CHPID via the switch....318
Scenario IV – Configuration changes for installing a FICON bridge card in a 9032–5 ESCON director....320
Scenario V – Establish a PPRC connection consisting of several PPRC links....325

Chapter 13. HCD-only tasks....331
Defining and viewing configuration data....331
Adding similar objects to a configuration....331
Copying configurations and merging IODFs....331
Transferring partition configurations....331
Upgrade an IODF from an earlier version of HCD....331
Querying supported hardware and installed UIMs....331
Viewing and printing the activity log....331
Support of large IODFs....331
Activating and processing configuration data....332
Building IOCDSs....332
Managing IPL attributes....332
Dynamic activate....332
Build configuration data sets....332
Building and processing CONFIGxx members....332

Chapter 14. Importing and exporting data....333
Data format for importing and exporting....333
File structure....334
Record structure....334
Column formats....335
Hierarchical object keys....337
Unique key....337
Future compatibility and extensibility....337
Working with the import and export function....337
Importing data....337
Exporting data....344
Importing in stand-alone mode....345
Exporting in stand-alone mode....345
Using import as a migration aid....345
Table summary....347
Detailed table formats....349
PROC table....349
PART table....351
CHPID table....352
DIR table....354
PORT table....354
CONTRLLR table....355
Controller channel interface (CCI) table....357

Controller device interface (CDI) table....358
CU table....359
STRING table....360
String interface (SI) table....362
UNIT table....363
DEVICE table....363
XBAR table....364
XBARCFG table....365
Crossbar channel interface (XCI) table....366
Crossbar control unit interface (XCUI) table....366
CABINET table....366
PANEL table....367
PATCHPRT table....367
CONVERTR table....368
CABLE table....369
LOGLINK table....370
CONNECT table....373
PPRCLINK table....376
DEVCU table (export-only)....377
CUCHPID table (export-only)....377
DEVIMAGE table (export-only)....378

Chapter 15. Physical description files (PDF)....379
Loading PDFs....379
Customizing PDFs....379
PDF syntax....379
Controller PDF syntax....380
String PDF syntax....382

Chapter 16. Problem determination....385
Identifying and handling problems....385
Product identifiers....386
Application error....386
Controlled exit at the workstation....386
Communication problems using TCP/IP under z/OS....386
Communication problems using TCP/IP under z/VM....387
Message problems....387
Incorrect results....387
Incorrect configuration diagram....388
Problems printing reports/configuration diagrams....388
Abnormal termination of HCD host....388
HCD host problems....389
Problems when using the RMF DDS....389
Problems with HCM User's Guide....390
Exporting an MCF....390
Incorrect help information....390
Using the HCD trace facility....391
Using the z/VM HCD TCP/IP Dispatcher for HCM trace facility....391
z/VM HCD TCP/IP Dispatcher for HCM commands....391
The HCMERR.LOG PC file....392
Searching problem reporting databases and reporting problems....393
Reporting problems to IBM....393

Chapter 17. HCM messages....395

Appendix A. Alternate processor definitions....415
Managing shadow processors during an HCM session....415

Creating a new processor under a machine.... 415
Deleting a processor from a machine.... 415
Changing the processor style.... 415
Making a processor primary or alternate.... 415
Assigning processors to machines during IODF load/resync.... 416

Appendix B. One Processor Per Partition (OPPP) machines.... 417
IODF load and resync.... 417
Creating and deleting OPPP machines.... 417
Editing OPPP processors.... 417
CHPIDs, channels, and CHPID mapping.... 418
Connecting OPPP machines.... 420
CTC connections to OPPP machines.... 420

Appendix C. Naming conventions in dialogs and reports.... 421
Cable descriptions.... 423

Appendix D. Menu bar choices summary.... 425
File menu.... 425
Edit menu.... 426
View menu.... 427
Locate menu.... 428
Create menu.... 429
Utilities.... 430
Options menu.... 431
Help menu.... 431

Appendix E. Special setup considerations for z/OS.... 433
HCM client: configurable login parameters.... 433
Host messages.... 434
HCM Client: TCP/IP related messages.... 435

Appendix F. Accessibility.... 437
Accessibility features.... 437
Consult assistive technologies.... 437
Keyboard navigation of the user interface.... 437
Dotted decimal syntax diagrams.... 437

Notices.... 441
Terms and conditions for product documentation.... 442
IBM Online Privacy Statement.... 443
Policy for unsupported hardware.... 443
Minimum supported hardware.... 443

Trademarks.... 445

Accessing licensed books on the web (z/OS).... 447

Accessing licensed books on the web (z/VM).... 449
Glossary.... 451

Index.... 463

Figures

1. Relationship between HCM Client, HCM Dispatcher, and HCM Agent..... 7

2. Sample Job for Trace Data Set Allocation..... 10

3. Sample AT-TLS policy for secure communication setup for the host:..... 21

4. Login dialog..... 25

5. Customize Field Names..... 30

6. Adding User data into customized user fields..... 31

7. HCD Error Message..... 31

8. Message Details dialog..... 32

9. Message List Dialog..... 32

10. HCM Message..... 33

11. Change View Attributes Dialog..... 34

12. Comparison of horizontal and vertical processor view..... 37

13. Context menu of a spanned CHPID in a vertical processor view..... 37

14. Highlighting an Object, without Visualized Configuration..... 39

15. Highlighting an Object, with Visualized Configuration..... 40

16. Selection List for Locating Objects, in this Example: Locating Devices..... 44

17. Filter dialog..... 45

18. Build New Processor Order for the Diagram..... 47

19. Configuration Files and IODFs..... 50

20. Confirm Database Resynchronization dialog..... 54

21. After loading the IODF in which control units have no or different serial numbers..... 58

22. After loading the IODF in which control units have the same serial numbers..... 58

23. High-level Qualifier dialog..... 59

24. IODFs dialog....60
25. Choose Local File dialog....61
26. IODF Details....62
27. IODFs dialog for z/VM....64
28. Machines dialog....65
29. File Open dialog....68
30. Confirm Database Upgrade....70
31. CHPID Mapping Tool Support....73
32. Export IODF....75
33. Import IODF....75
34. Check Configuration File....76
35. Save Production File dialog....77
36. Activity Log dialog....79
37. Set Access Mode....81
38. Physical Mismatch Resolution....86
39. Configuration Packages....89
40. Create Configuration Package....90
41. Configuration Package Objects....91
42. Add OS configuration objects....91
43. Transmit a configuration package....92
44. Specify Import JCL....93
45. Message EEQ0212E....96
46. VOLSER Reconciliation dialog....97
47. Wizard page flow and work flow....100
48. Processor dialog for SMP processors....107

49. Processor dialog for XMP processors....108
50. Create Processor....109
51. PCIe Functions dialog....111
52. Add PCIe Function dialog....112
53. Create Channel Subsystem....113
54. Partitions....115
55. CHPIDs....118
56. Choose Columns....119
57. CHPID Operation Mode Symbols....121
58. CHPID....122
59. Create CHPIDs....123
60. Edit IQD Channel Parameters....124
61. Allow for more than 160 TCP/IP stacks....124
62. Specify HCA attributes....125
63. Specify Coupling Port attribute....125
64. Assign Partitions....127
65. Spanned CHPIDs in the Configuration Diagram....129
66. Physical connection information for spanned CHPIDs....129
67. Controller....131
68. Edit Controller....133
69. Arrange Controller....134
70. Arrange Control Units and Connections....136
71. Group Controllers....137
72. Interface Attribute....138
73. Control Unit List....139

74. Add Existing Control Unit to Internal Controller..... 140
75. String..... 141
76. Create String..... 141
77. Physical Descriptions dialog..... 142
78. Edit String..... 143
79. Units/Devices..... 144
80. Edit Unit..... 146
81. I/O Device List..... 147
82. Create I/O Devices..... 148
83. Selecting processors for changing the subchannel set placement for devices..... 149
84. Edit Subchannel Set ID..... 150
85. Converters..... 151
86. Create Converter..... 152
87. Edit Converter..... 153
88. Replace Converter..... 154
89. Switch..... 156
90. Edit Switch..... 156
91. Port Attributes..... 157
92. Configuration for Switch XX..... 158
93. Configuration Port Matrix..... 159
94. Configuration Port Matrix - Filtered and CHPIDs on Top..... 161
95. Edit Crossbar Switch..... 163
96. Crossbar Configurations..... 164
97. Cabinet..... 168
98. Create General Box..... 172

99. Positioning an Object for Daisy-Chained Connections..... 177

100. Select Controller Device Interface..... 179

101. Reachable Control Units..... 180

102. Affected I/O Devices..... 181

103. I/O Device <=> Processor Definitions..... 181

104. Edit I/O Device <=> Processor Definitions..... 182

105. Select Connection Target(s)..... 184

106. Affected Control Units..... 186

107. Processor and CHPID Connections..... 186

108. Processor and CHPID Connections..... 188

109. Processor and CHPID Connections..... 188

110. Choose Channel Subsystems..... 189

111. Edit Control Unit <=> Processor Definitions..... 190

112. Add Control Unit <=> Processor Definitions..... 191

113. Switch..... 194

114. Select Multiple Connection Target(s)..... 194

115. CHPID's Dynamic Switch Modification..... 195

116. CTC connection between two CHPIDs on different processors through a switch..... 196

117. Create CTC Connection..... 196

118. CTC Connections..... 197

119. Edit CTC Connection..... 198

120. View CTC Connection Details..... 199

121. View Unused CTC Control Units, Devices, and CHPID Connections..... 201

122. Coupling Facility Connection..... 204

123. Coupling Facility Connections..... 205

124. Create Coupling Facility Connection..... 205
125. Choose CU and Device Numbers..... 206
126. Cable Assignments Dialog..... 208
127. Assign Cable Dialog..... 209
128. Select Cable Labels to Print Dialog..... 211
129. Choose a Cable Label Format to Print Dialog..... 211
130. Cabinet Dialog..... 213
131. Select Connection Target(s)..... 214
132. Edit Fiber Link..... 215
133. Configuration Diagram..... 217
134. Disconnect Dialog..... 218
135. Autoconfiguration - Options..... 223
136. Discovered Controller List..... 223
137. Proposed Control Unit List..... 224
138. Edit Proposed Control Unit..... 225
139. Proposed Device List..... 225
140. Edit Proposed Devices..... 226
141. Map Proposed Control Units..... 226
142. Map Proposed Devices..... 227
143. Autoconfiguration - Summary (excerpt)..... 228
144. Autoconfiguration Policies..... 229
145. Autoconfiguration Policies - LP Groups..... 229
146. Add LP Group..... 230
147. Autoconfiguration Policies - OS Groups..... 230
148. Edit OS Group..... 231

149. Example for Implied Ports....232
150. Export IOCP or I/O Configuration Statements....236
151. Import I/O Configuration Statements....237
152. Available performance data for a processor....239
153. Total utilization of the processors by partitions....240
154. Partition performance data....241
155. Partition related performance data....241
156. Specify RMF resource type....242
157. WWPN Prediction Tool Support....242
158. Operating System Configurations dialog....246
159. Create Operating System Configuration....246
160. List Devices for OS Configuration dialog....248
161. Add I/O Devices to OS Configuration dialog....249
162. Define Device <=> OS Configuration Parameters dialog....250
163. List Device <=> OS Configuration Definitions dialog....251
164. Include Devices in Esoterics dialog....252
165. List Consoles dialog....253
166. Add Console dialog....254
167. Edit Console dialog....254
168. Create Eligible Device Table dialog....255
169. Create Esoteric Device Group dialog....256
170. List Devices for Esoteric dialog....257
171. Add I/O Devices to Esoteric dialog....258
172. List System-Defined Generics dialog....259
173. List User-Defined Generic dialog....260

174. Printing a configuration diagram..... 264

175. Generate Reports.....266

176. Printer format sample.....267

177. XML/HTML format sample.....268

178. CSV format sample..... 269

179. IODF Reports..... 279

180. IODF Compare Reports..... 280

181. IODF Compare CSS/OS Reports..... 281

182. HCM Compare Reports..... 282

183. HCM Compare Reports - Options..... 282

184. Create PPRC Connection (1).....326

185. Create PPRC Connection (2).....327

186. Create PPRC Connection (3).....328

187. Create PPRC Connection (4).....329

188. Hardware Configuration Manager Confirmation Message.....338

189. Choose HCM File for Import Dialog.....339

190. File Open Dialog.....339

191. Choose IODF Dialog.....340

192. Choose Import Files Dialog..... 341

193. Text File Options dialog..... 341

194. Proposed Description Updates Dialog..... 342

195. Export Data Dialog..... 344

196. CHPID <> Channel Mapping Dialog.....419

Tables

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

2. Required Minidisks 11

3. Required Minidisks for Installation 12

4. Optional Minidisks 12

5. Quick File Reference 13

6. Configuration diagram connections 35

7. IODF Load versus IODF Synchronize 53

8. Opening an existing configuration file with a new IODF 69

9. I/O configuration: responsibilities and tasks 82

10. I/O configuration: required access rights and tasks 82

11. Printing of reports 96

12. Symbols in port matrix 159

13. Table summary 347

14. PROC Table 349

15. PART Table 351

16. CHPID Table 352

17. DIR Table 354

18. PORT Table 354

19. CONTRLLR Table 355

20. CCI Table 357

21. CDI Table 359

22. CU Table 359

23. STRING Table 360

24. SI Table..... 362
25. UNIT Table..... 363
26. DEVICE Table..... 363
27. XBAR Table..... 365
28. XBARCFG Table..... 365
29. XCI Table..... 366
30. XCUI table..... 366
31. CABINET Table (also containing general boxes)..... 366
32. PANEL Table..... 367
33. PATCHPRT Table..... 367
34. CONVERTR Table..... 368
35. CABLE Table..... 369
36. LOGLINK Table..... 370
37. CONNECT Table..... 373
38. PPRCLINK Table..... 376
39. DEVCU Table (export-only)..... 377
40. CUCHPID Table (export-only)..... 377
41. DEVIMAGE Table (export-only)..... 378
42. Controller PDF attributes..... 381
43. String PDF Attributes..... 382
44. DDS problem handling (1)..... 389
45. DDS problem handling (2)..... 390

About this document

This guide explains how to use **Hardware Configuration Manager** (**HCM**), a comprehensive, graphical hardware configuration and operations tool. HCM is a PC-based client/server interface to z/OS Hardware Configuration Definition (HCD) that combines the logical and physical aspects of hardware configuration management. 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 a z/VM® system has basic knowledge of z/OS and z/VM and hardware configuration.

For information about configuration planning for z/OS refer to *z/OS HCD Planning*. For information about 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 25
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 49
describes the relationship between HCM and HCD (IODFs, configuration files); explains physical and logical objects; provides information about how to create a configuration from an IODF or from scratch; provides information about how to open an existing configuration and how to save, close or copy the configuration.

Chapter 5, "Stand-alone mode," on page 95
explains how to invoke and work with the stand-alone mode.

Chapter 6, "Modifying objects in your configuration," on page 99
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 175
explains how to connect and define logical objects.

Chapter 8, "Physical connections," on page 207
explains how to add purely physical objects to existing connections: cables, patchports, and crossbar switches.

Chapter 9, "Utilities," on page 221
describes utilities HCM provides to make complex configuration tasks easier.

Chapter 10, "Operating system configurations," on page 245
explains how to specify OS configurations, EDTs, esoteric device groups, and generic device groups, as well as how to define I/O devices to the OS configuration.

© Copyright IBM Corp. 1995, 2021

Chapter 11, “Reports and diagrams,” on page 263
describes how to print various reports, context diagrams and configuration diagrams from HCM, and gives examples.

Chapter 12, “Scenarios,” on page 285
contains a number of configuration scenarios, step-by-step, for you to follow.

Chapter 13, “HCD-only tasks,” on page 331
lists the tasks which cannot be carried out using HCM.

Chapter 14, “Importing and exporting data,” on page 333
explains how to exchange data between your processes, application, and data bases and the configuration data of HCM via ASCII files.

Chapter 15, “Physical description files (PDF),” on page 379
describes the different PDFs and their syntax.

Chapter 16, “Problem determination,” on page 385
explains how to diagnose problems that may arise using HCM.

Chapter 17, “HCM messages,” on page 395
contains all HCM messages.

Appendix A, “Alternate processor definitions,” on page 415
explains how HCM handles multiple logical processors in one physical machine.

Appendix B, “One Processor Per Partition (OPPP) machines,” on page 417
explains how HCM handles machines not represented in IBM's processor partitioning scheme.

Appendix C, “Naming conventions in dialogs and reports,” on page 421
shows how objects in HCM are typically described.

Appendix D, “Menu bar choices summary,” on page 425
lists all commands that can be used in the different menus.

Appendix E, “Special setup considerations for z/OS,” on page 433
discusses special considerations related to set up the communication between HCM an d z/OS.

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*.

To view, search, and print z/OS publications, go to the z/OS Internet library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).

Softcopy documentation is available as online collection kit that is available in compressed format for download from the IBM publication center.

Also visit the HCD and HCM home page (www.ibm.com/systems/z/os/zos/features/hcm), which provides information concerning product updates, newsletters, conferences, and more. For more information about **z/OS HCD**, 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 might be useful:

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

Title	Order Number
z/VM: I/O Configuration This book contains detailed information about z/VM HCD.	SC24–6100
z/VM CP Command and Utility Reference	SC24–5967

Table 1. Related z/VM I/O Configuration References (continued)	
Title	**Order Number**
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	SC34-2668
z/OS MVS Device Validation Support	SA38-0697
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 (www.vm.ibm.com/library).

How to send your comments to IBM

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

Important: If your comment regards a technical question or problem, see instead “If you have a technical problem” on page xxvii.

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function
If your comment or question is about z/OS itself, submit a request through the IBM RFE Community (www.ibm.com/developerworks/rfe/).

Feedback on IBM Documentation
If your comment or question is about the IBM Documentation functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Documentation at ibmdoc@us.ibm.com.

Feedback on the z/OS product documentation and content
If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The following deliverable title and order number: z/OS and z/VM HCM User's Guide, SC34-2670-50
- The section title of the specific information to which your comment relates
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive authority to use or distribute the comments in any way appropriate without incurring any obligation to you.

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

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

© Copyright IBM Corp. 1995, 2021

Summary of changes

Summary of changes for z/OS Version 2 Release 5 (V2R5)

The following changes are made for z/OS Version 2 Release 5 (V2R5).

New

September 2021

There is no new content in this release.

Changed

September 2021

- All references to Windows was changed to provide support only to Windows 10.

Deleted

September 2021

- All releases of Windows prior to Windows 10 were deleted.

Summary of changes for z/OS Version 2 Release 4 (V2R4)

The following changes are made for z/OS Version 2 Release 4 (V2R4).

Hardware support

Partition Enhancements:

Support for new partition type for the processor types 3906 and 3907.

1. The new partition type is called FW.
2. The new partition type is only supported by partition number B in the highest channel subsystem of the processor.
3. The name of MCS firmware partition is required to be MCS_1.

Summary of changes for z/OS Version 2 Release 3 (V2R3)

The following changes are made for z/OS Version 2 Release 3 (V2R3).

Hardware support

Processors supported:

HCM supports a new processor family (processor types 3906-M01, -M02, -M03, -M04, -M05) and a new processor family (processor type 3906-LM1, -LM2, -LM3, -LM4, -LM5).

PCIe function enhancements:

Two new PCIe function types ZHYPERLINK and ROCE-2 are supported on the new processor family. The PCIe functions of type ZHYPERLINK and ROCE-2 require a PORT value.

New CHPID type enhancement:

HCM supports a new channel path (CHPID) type called CL5. The CL5 adapter is defined with a physical channel ID (PCHID) value and a PORT value.

© Copyright IBM Corp. 1995, 2021

Changes made in z/VM Version 6 Release 2

Summary of changes for SC33–7989–13 issued December 2011

HCM for z/VM V6R2 includes the new functions available with HCM for z/OS.

The following functions of previous HCM releases are also made available for z/VM V6R2:

- HCM for z/VM supports the Internet Protocol, Version 6 (IPv6), which can be used when establishing a communication session with z/VM.
- Users of HCM for z/VM can now use the dialogs for working with configuration packages as described in "How to work with configuration packages" on page 88.

Note: I/O Autoconfiguration is not available for z/VM.

Changes made in z/VM Version 6 Release 1

HCM for z/VM V6R1 includes all functions available with HCM for z/OS V1R10 and also provides support of the WWPN Prediction Tool (see "Using the WWPN Prediction Tool Support" on page 242).

z/VM V6R1 supports the zEnterprise® 196 (z196) processor family (processor types 2817-M15, -M32, -M49, -M66, -M80), and also supports the new OSM and OSX channel paths and a third subchannel set with ID 2 for these processors.

Note that HCM's handling of MUA enabled IODFs is only available on z/OS.

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 with multiple channel subsystems, 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 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 PC-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 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 more information about how HCM and HCD work together, see Chapter 4, "Building your configuration," on page 49.

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 the wanted 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 intelligently adjusts 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 highlights 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 13, "HCD-only tasks," on page 331).

© Copyright IBM Corp. 1995, 2021

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 objects in the configuration of a system. 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 *HCD and HCM for z/VM Program Directory* for z/VM.

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 4. Before you start this process, ensure that you meet the requirements as stated in "Prerequisites" on page 3.

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

   Depending on the operating system, follow the steps described in "Setting up TCP/IP definitions for z/OS" on page 6 or "Setting up TCP/IP definitions for z/VM" on page 10.

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
- "Uninstalling HCM" on page 23

Prerequisites

The following requirements must be met to run HCM.

Workstation hardware requirements

Hardware requirements to run HCM effectively are:

- About 200 MB of free disk space
- Color display (1024 x 768 resolution is required)
- network adapter

Processing large configurations might require additional disk space and may benefit from additional memory.

Software requirements

The following software requirements must be met to run HCM.

Workstation

On your workstation, you need the following:

Operating system

You need to run on the following operating system:

- Windows 10

HCM installation

You need a method to download the code from the host to the workstation (for example, FTP or Personal Communications).

For Windows 10, you need to have administrator rights to install HCM.

© Copyright IBM Corp. 1995, 2021

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 TCP/IP.

For PTFs that need to be installed for the current z/OS and z/VM operating system releases, see the PSP-bucket or refer to the HCM readme file readme.rtf.

Communication

UNIX System Services and TCP/IP must be running and the HCM user ID must be entitled to access the UNIX System Services. **Note:** Superuser authority is not necessary, but a home directory must be provided for the user.

Browsing performance data

To access RMF Monitor III performance data for a selected object from an HCM configuration diagram, you must have RMF installed with the RMF Distributed Data Server (DDS) set up and running. Since the DDS is only available on z/OS, you cannot access performance data from z/VM.

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 MSI installation package on your workstation:

1. The HCM MSI installation package is available
   - in the data set member SYS1.SEEQINST(EEQINSTM) on the z/OS host
   - or in file EEQINSTM MSIBIN on the z/VM host from minidisk CBDIODSP 400.

   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.

   Download the HCM MSI package in *binary* mode onto your workstation into a temporary folder and rename it to `'eeqinst.msi'`.

   Note: If you want to install HCM V1R11 or later for the first time, with currently having an HCM version V1R10 or earlier installed, you first must uninstall your current HCM version.

   Your current settings in the HCM INI file are not deleted during uninstall, but are used for the new installation. See also “Uninstalling HCM” on page 23.
2. Distinguish the following cases:
   - If you do not have any HCM version installed, then install the HCM MSI package using the Windows Installer by double-clicking on the package file or by issuing the command:

     ```
     msiexec /package eeqinst.msi
     ```
   - If you already have HCM V1R11 or later installed, you can either start the installation directly in a Windows command prompt with the command:

     ```
     msiexec /i eeqinst.msi REINSTALLMODE=voums REINSTALL=ALL
     ```

     or you must uninstall it before starting the new installation.
3. The Windows Installer guides you through the installation. During the installation, you are asked for the destination location of HCM. You can choose the default folder:

```
C:\Program Files\IBM\Hardware Configuration Manager\
```

or specify a different folder by clicking the **Change** button.

Note: If you install HCM into the default folder and want to use your own physical description files (PDF), you must have administrator rights to access the default folder. Otherwise, you must install HCM into a different folder where you have write access.

4. The HCM installation creates a program folder ***IBM Hardware Configuration Manager*** in your **Start** menu from where you can launch HCM. Once HCM is successfully installed on your system, you can delete the HCM MSI package.

Purpose and location of the HCM INI file

The HCM INI file (EEQHCM.INI) is created after the first successful login to HCM. HCM uses this file to store information and user-specific settings across HCM sessions, for example, column settings in dialogs or folder names.

In HCM releases prior to z/OS V1R11 HCM, this HCM INI file was stored in the C:\WINDOWS folder. Starting with z/OS V1R11 HCM, the INI file is located in the user-specific application data folder. This folder depends on the Windows operating system, as shown in the following example:

Default folder for Windows 10 for user `usry`:

```
C:\Users\usry\AppData\Roaming\IBM\Hardware Configuration Manager
```

Thus you can store multiple user-specific instances of the EEQHCM.INI file on one workstation.

During installation, HCM checks if an INI file exists in the user-specific folder. If HCM finds an EEQHCM.INI file there, it keeps it and uses it for all subsequent HCM sessions. Otherwise, if no INI file is found, HCM looks for an existing copy of this file in the C:\WINDOWS folder. If one exists there, HCM copies this file to the new location and keeps the old version in its original location. If no INI file instance is found, a new one is created. All future updates are done in the new user-specific file.

Extracts from an example INI file:

```
[HCM]
…
CONFGDIR=D:\hcmcfg
…
TCPIPHOST=hostname.company.com
TCPIPPORT=51107
…
```

Note:

1. 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 defaults are usually sufficient. The problem might be the TCP/IP configuration/customization (LE environment, DNS name server, and so on. First check the reason for the timeout before increasing the defaults. See Chapter 16, “Problem determination,” on page 385 for more details.
2. There might be situations when you need an INI file for the first login request, for example, with very slow JES queues. In this case, you can simply use an editor and create the INI file manually with the desired contents.

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 does not help, obtain the current host PTF once more and install it. Download the files again and retry the installation. If this still does not solve your problem, contact IBM.

Installing 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 the complete product with the new updates as well as all updates from earlier service levels. That is, you only have to install the newest service level to obtain all product updates.

During installation, HCM displays the contents of the README file, which contains information about the current enhancements. This file is called `readme.rtf` and resides in the installation folder after you have installed the service level. Always see the `readme.rtf` file before you install or use the new service level.

For information about service levels, see the HCD and HCM home page (www.ibm.com/systems/z/os/zos/features/hcm).

Installing HCM in a LAN environment

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

When you keep the HCM product folder 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.

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 PC from which HCM has been installed. If you want to use HCM from the LAN resource on a different PC, you must create an icon on your desktop or in your **Start** menu to work with HCM.
- If a controlled exit occurs, or if some logging mechanism is activated for debugging purposes, HCM writes some log files to the installation folder. If in your environment the HCM installation folder on the LAN is write protected, the log files could not be created. Additionally, if you want each LAN user to keep own log files, you can specify a folder other than the default one, using the ERRLOGDIR parameter in the [HCM] section of the EEQHCM.INI file. See “The HCMERR.LOG PC file” on page 392 for more information about 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 PC from which it has been installed originally.

Establishing the host communication

After you have successfully installed HCM on your workstation, you must link the host and the workstation depending on your operating system:

- “Setting up TCP/IP definitions for z/OS” on page 6
- “Setting up TCP/IP definitions for z/VM” on page 10

Setting up TCP/IP definitions for z/OS

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 - called 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.

Note: Starting with z/OS V1R11, HCM supports the Internet Protocol IPv6 in addition to IPv4 to establish a communication session with the HCD host.

The following picture provides an overview of the structure and illustrates the relations between the HCM client, HCM dispatcher, and the HCM agent.

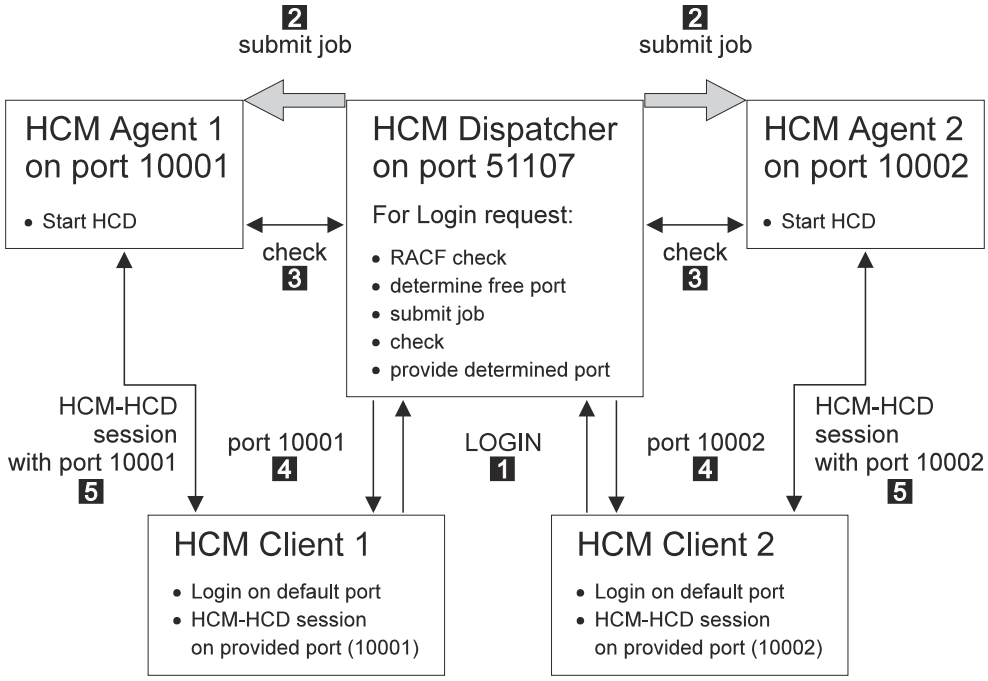


Figure 1. Relationship between HCM Client, HCM Dispatcher, and HCM Agent

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. As soon as 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 is 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 8). 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 8 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 are 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 z/OS.

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
It is possible to determine a different range for the ports chosen by the HCM dispatcher for communication with the HCM agent during start of the dispatcher. The port range used must be configured to be available and permitted to the HCM agent. Set the variables P0 and P1 to appropriate values. Note that 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).

Sample JCL for creating a user ID for UNIX System Services

```
//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(*)
/*
```

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 2. 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 about how to do that.

To verify that TCP/IP is enabled, try one of the the following commands from the Windows command prompt:

- `ipconfig /all` - this displays your Windows IP configuration.
- `ping` ***`hostaddress`*** - where ***`hostaddress`*** is the IP name or address of your z/OS host where the HCM dispatcher runs.

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 about 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 topics:

- "Setting up the service virtual machine" on page 11
- "Setting up the z/VM HCD TCP/IP Dispatcher for HCM" on page 13

How to invoke HCM as the interface to z/VM HCD is described in "Starting and stopping HCM for z/VM" on page 26.

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" on page 11
- "CP directory entry for service virtual machine" on page 12

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 2. Required Minidisks

Minidisk	Description
CBDIODSP 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 3. Required Minidisks for Installation	
Minidisk	**Description**
6VMHCD20 2C2	z/VM HCD sample file and user local modification disk
MAINT 193	Production disk containing Reusable Server Kernel (RSK) library files

Table 4. 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 11. 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> 256M 999M BCEFG
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.

256M means that the virtual machine has 256M of primary address space available after logging in. This value is recommended if large IODFs have to be processed.

999M means that the user on the virtual machine is allowed to enlarge the primary address space size up to 999M (using the CP command DEFINE STORAGE).

BCEFG means that the user ID CBDIODSP is allowed to enter CP commands of privilege classes B, C, E, 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 17 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. IODFs in VM must reside on a minidisk that is formatted with 4 K blocksize.

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

After the previous preparations have been 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 14
- “Setting up Security” on page 17

Table 5 on page 13 provides an overview of files that you need to prepare for this purpose. For more detailed information see “How to set up required files” on page 14.

Table 5. Quick File Reference

File Name	Description
PROFILE EXEC	A REXX EXEC which is executed upon login to the z/VM HCD TCP/IP dispatcher.
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.

Table 5. Quick File Reference (continued)	
File Name	**Description**
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.
Authorization files	Files used to hold authorization data managed by the RSK. The RSK will create these files if they do not exist.
PROFILE RSK	REXX EXEC which is executed when the z/VM HCD TCP/IP dispatcher is started.
CBDCONF NAMES	z/VM HCD names file defining default file names and settings.
CBDSDISP EXEC	REXX EXEC used to start the z/VM HCD TCP/IP dispatcher.
z/VM HCD Profile (HCD PROFILE)	Profile defining the settings of your z/VM HCD session.

How to set up required files

This section handles the required files as introduced in Table 5 on page 13, 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 SAMPPROF` 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 about 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 can 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 are 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 use 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 about 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 1:

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 about 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 about 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 2:

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" on page 17
- "How to set up security using an external security manager" on page 17: 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 391) 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 12. 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 19.

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
```

Client Authentication

Note:

In the descriptions that follow, it is assumed that you use the following routines or programs:

- the CSL routine DMSPASS provided by IBM
- RACF/VM as external security manager (ESM)
- RACROUTE provided by IBM as the request handler

If `:SECURITY.ESM` is specified in CBDCONF NAMES, and an HCM client requests access to z/VM HCD, the z/VM HCD TCP/IP dispatcher will call DMSPASS to ensure that the provided user ID and password 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.

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 about 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. Enable HCD/HCM to access DIAGNOSE X‘88’:

   You must enable the CBDIODSP service machine for DIAGNOSE X‘88’ access. If RACF is being used to control DIAGNOSE X‘88’ access, enable DIAGNOSE X‘88’ access for CBDIODSP by completing the following steps:

   Step 1. Enable RACF/VM profile protection for DIAGNOSE X‘88’:

   a. Confirm that there are no members called DIAG088/NOCTL in the active VMXEVENT profile.
   b. Create a profile called DIAG088 in the VMCMD class with a default access of NONE:

      ```
      RDEFINE VMCMD DIAG088 UACC(NONE)
      ```

   c. Ensure that the VMCMD class is active:

      ```
      SETROPTS CLASSACT(VMCMD)
      ```

   Note: If you do not enable RACF profile protection, the CBDIODSP server must be defined with OPTION DIAG88 in its directory entry.

   Step 2. Give the CBDIODSP server permission to perform password validation using DMSPASS (which uses DIAGNOSE X‘88’ subcode 8):

   ```
   PERMIT DIAG088 CLASS(VMCMD) ID(CBDIODSP) ACCESS(READ)
   ```

   For more information, see the *z/OS Security Server RACF Security Administrator's Guide*

   .

3. 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.

4. 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)`.

5. In your PROFILE EXEC (see page "How to set up required files" on page 14), initialize the connection to the RACF/VM virtual machine by removing the comment characters in the following line:

```
/* RPIUCMS INIT */
```

6. To access the HCD services remotely on z/VM, a user needs to logon to the HCD dispatcher. The logon is done with a user ID and password as defined to the external security manager, for example, IBM Security Server RACF. The user must have the same access rights as for using HCD directly in the operating system.

   If the APPL class for the security product is active, a profile can be defined to allow only certain users to log on to the HCD Dispatcher. You can manage access to the HCD application by profile CBDSERVE in the APPL class. Users who are allowed to use HCD need READ access to this profile. Sample definitions for user HCDUSER for RACF would look like:

```
RDEFINE APPL CBDSERVE UACC(NONE)
PERMIT CBDSERVE CLASS(APPL) ID(HCDUSER) ACCESS(READ)
```

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, 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).

Encrypting the host communication for z/OS

After you have successfully set up TCP/IP definitions for your operating system, it is recommended to encrypt the host communication. You can follow the steps:

- Set up the secure communication for the host.
- Set up the secure communication for the workstation.

However, if you do not want to encrypt the host communication, you can skip these steps.

Note: Encrypting the host communication is only available on z/OS.

Set up secure communication for the host

This topic describes how to enable secure communication with the host using Application Transparent - Transport Layer Security (AT-TLS).

Note: Using AT-TLS to secure communication with the host affects all HCM clients. If AT-TLS is used to enable secure communication with the host, all HCM clients must use TLS to protect communication with the host. Unprotected connections to the HCM server will no longer be allowed.

Before you can start with the HCM server setup, you must enable the z/OS Communications Server Policy Agent for AT-TLS. Information about setting up AT-TLS communication is provided in the *z/OS Communications Server: IP Configuration Guide*.

The following example uses a RACF key ring. The key ring must contain a valid default certificate. The user ID who starts the dispatcher program must have access to the key ring. Consider using a certificate from a trusted Certificate Authority (CA). For more information on CA, see the *z/OS Security Server RACF Security Administrator's Guide*.

For other security management products, refer to your security product documentation.

```
# HCD Dispatcher as the server rule

TTLSRule                                    HCD_Dispatcher
{
    LocalPortRange                          51107
    Direction                               Inbound
    Priority                                20
    TTLSGroupActionRef                      grp_Production
    TTLSEnvironmentActionRef                HCD_Disp
}

TTLSEnvironmentAction                       HCD_Disp
{
    HandshakeRole                           Server
    TTLSKeyRingParms
    {
        Keyring                             <your key ring name>
    }
    TTLSEnvironmentAdvancedParmsRef         HCD_EnvAdvParms
}

# The agents as servers rule

TTLSRule                                    HCD_Agents
{
    LocalPortRange                          10000-10010    # agents ports range

    Direction                               Inbound
    Priority                                20
    TTLSGroupActionRef                      grp_Production  # enable TTLS
    TTLSEnvironmentActionRef                HCD_Agt
}

TTLSEnvironmentAction                       HCD_Agt
{
    HandshakeRole                           Server
    TTLSKeyRingParms
    {
       Keyring                              <your keyring name>
    }
TTLSEnvironmentAdvancedParmsRef             HCD_EnvAdvParms
}

# The dispatcher as the client rule

TTLSRule                                    HCD_Dispatcher_As_Client
{
    RemotePortRange                         10000-10010 # Agent server port range
    Direction                               Outbound
    Priority                                20
    TTLSGroupActionRef                      grp_Production # enable TTLS
    TTLSEnvironmentActionRef                HCD_Disp_As_Client
}
TTLSEnvironmentAction                       HCD_Disp_As_Client
{
    HandshakeRole                           Client
    TTLSKeyRingParms
    {
       Keyring                              <your keyring name>
    }
TTLSEnvironmentAdvancedParmsRef             HCD_EnvAdvParms
}

# Common Production Group that all Rules use

TTLSGroupAction                             grp_Production
{
    TTLSEnabled                             On
    Trace                                   2
}

TTLSEnvironmentAdvancedParms                HCD_EnvAdvParms
{
    SSLV2                                   Off
    SSLV3                                   Off
    TLSV1                                   Off
    TLSV1.1                                 On
    TLSv1.2                                 On
}
```

Figure 3. Sample AT-TLS policy for secure communication setup for the host:

After you have completed your Policy Agent configuration, use the REFRESH command to refresh the Policy Agent configuration.

```
MODIFY PAGENT,REFRESH
```

TTLSRule: LocalPortRange

For server rules, this is the local port the application is listening on. 51107 is the listening port of the dispatcher and the agents listen on ports 10000-10010. For more information, see “Starting the HCM dispatcher as a started task” on page 8 for the Port Range definition.

TTLSRule: Direction

Specifies the direction the connection must be initiated from for this rule's action to be performed. In this example, Inbound is specified for server rules, which means that the rule applies to connection requests that arrive inbound to the local host. Outbound is specified for client rules, which means the rule applies to connection requests initiated from the listening port of the dispatcher to the listening ports of the agents.

TTLSEnvironmentAction: HandshakeRole

Specifies the TLS handshake role to be taken for connections in this AT-TLS environment. In this example, Server is specified for the dispatcher's listening port, which means that the TLS handshake is performed as a server for inbound connections to that port. For outbound connections from the dispatcher to the agent ports, Client is specified for the handshake role which means that the TLS handshake is performed as a client.

TTLSKeyringParms: Keyring

Specifies the path and file name of the key database z/OS® UNIX file, the ring name of the SAF key ring, or the name of the z/OS PKCS #11 token. In this example, where <your key ring name> is the name of your SAF key ring.

TTLSGroupAction: TTLSEnabled

Indicates whether a selected connection should use AT-TLS security. For the dispatcher and agents, On is specified, which means the connections should use AT-TLS security.

TTLSEnvironmentAdvancedParms:SSL2
TTLSEnvironmentAdvancedParms:SSL3
TTLSEnvironmentAdvancedParms:TLSv1
TTLSEnvironmentAdvancedParms:TLSv1.1
TTLSEnvironmentAdvancedParms:TLSv1.2

Specify advanced attributes for an AT-TLS environment. HCM only supports TLSv1.1 and TLSv1.2. If On is specified for TLSv1.1 and TLSv1.2, the actual protocol version used will be negotiated to the highest version mutually supported by HCM and the host. Off should be specified for SSL2, SSL3 and TLSv1. It is very important to set the proper TLS level for your environment.

Set up secure communication for the workstation

The TLS protocol normally requires a server to supply a digital certificate to a client. The client will validate the server certificate by checking the trusted root CA in its own key data base. The client can then use the server's public key from the certificate during the rest of the TLS handshake. This process is called "TLS server authentication" and HCM requires its use. If the server authentication process fails, the TLS handshake is immediately terminated and the HCM connection fails.

Make sure that the server's root CA certificate is installed in Trusted Root Certification Authorities list on your workstation, so that Windows will automatically trust the host's certificate authority. If the root certificate is not installed in Trusted Root CA list, you can:

1. Obtain the root certificate that the host uses.
2. Follow Microsoft's documentation for the Microsoft Management Console (MMC) to:
    a. Enable the Certificates snap-in.
    b. Import the root certificate into Trusted Root CA.

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 about error messages, see Chapter 17, “HCM messages,” on page 395. You may also check for documentation updates in *HCM Documentation* under *Product Updates* provided on the HCD and HCM home page (www.ibm.com/systems/z/os/zos/features/hcm).

If you get messages that are not listed in Chapter 17, “HCM messages,” on page 395, inform IBM and provide the complete and exact message text (especially the internal description and the error stack information). If possible, make a screenshot of the message and provide information about the circumstances that caused the message.

Uninstalling HCM

Like most Windows applications, you can uninstall HCM using the standard Windows uninstall feature. Or, you can use the ***Uninstall*** function provided by HCM from the program folder ***IBM Hardware Configuration Manager***.

Note:

1. Uninstalling HCM only removes those folders and files that were originally installed by HCM. If you added files after the installation, for example physical description files (PDFs), these files and their associated folders are not removed. If desired, remove these files or folders manually. Also, your HCM configuration files are not deleted and you can keep them for later use.
2. The EEQHCM.INI file is not deleted when you uninstall HCM.

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 49).

Starting and stopping HCM for z/OS

Before invoking HCM, make sure that your TCP/IP connection to the HCD server on the host is running (see “Setting up TCP/IP definitions for z/OS” on page 6. 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** to display the **Login** dialog.

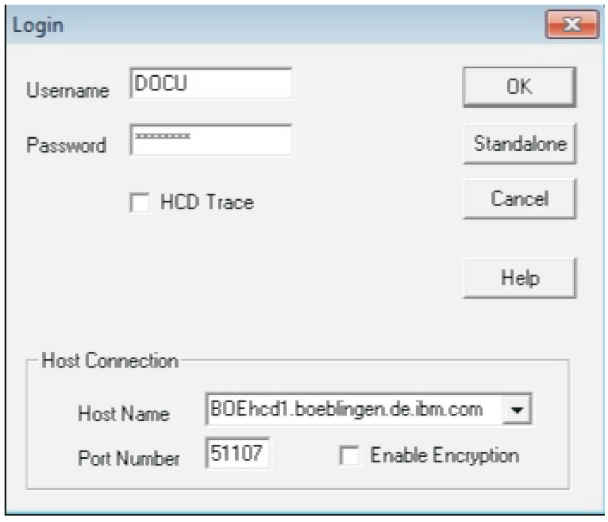


Figure 4. Login dialog

To login to HCM you must enter your user name and either a password or a password phrase, depending on the security setup on your host. A password phrase is a sentence (including blanks) of at least 9 and up to 100 mixed-case characters. HCM handles an input string of eight characters or less as a password and handles a string of nine characters or longer as a password phrase.

Note: Passwords and password phrases may be case sensitive. This depends on the security setup.

If you start HCM for the first time after installing it, you must enter a host name and a port number. In the ***Host Name*** entry field, HCM accepts symbolic host names or numeric IP addresses, which you can specify in either IPv4 or IPv6 format, depending on your network environment. The default port number is 51107. If your HCM dispatcher uses another port number, contact your system administrator.

HCM saves the entered values in the EEQHCM.INI file. Therefore, for subsequent log-ins, you only need to enter your password. This INI file is created *after* the first HCM session. It is located in a user-specific application data directory (see also “Purpose and location of the HCM INI file” on page 5).

© Copyright IBM Corp. 1995, 2021

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** terminates HCM.

If the login fails due to an expired password, you must logon to the host to change your password. You cannot change the password in your HCM session.

For secure connection to z/OS HCD, click in the **Enable Encryption** check box in the **Login** dialog to allow this HCM session to connect the HCD host using TLS. Make sure AT-TLS has been set up on the host in advance. See "Encrypting the host communication for z/OS" on page 19 for details on how to configure the host and the workstation. For the connection to z/VM HCD, remember to uncheck the **Enable Encryption** check box since encrypting the host communication is only available on z/OS.

For login troubleshooting, click in the **HCD Trace** check box in the **Login** dialog to enable the HCD host to trace 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 391 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 10. 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 6.2.1 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 25.
4. Enter user ID (default: CBDIODSP) and password of your service virtual machine (SVM) into the **Login** dialog (see also Figure 4 on page 25). 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 see:

```
SERVER STOP
RSK>
SERVER   BKWSRV0300I Shutdown initiated.
CBDD210I z/VM 6.2.1 HCD TCP/IP dispatcher for HCM stopped.
Ready;
```

Note:

Before you can use a final configuration on a z/VM system, make sure that the prerequisites described in chapter *Specify a Production IODF for IPL* in *z/VM: I/O Configuration* are satisfied.

Working with the user interface

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

Keyboard

Use the function keys and special keys as shortcuts for pull down choices. 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 most recently used named view saved with **Save Named View** from the **View** menu. When selected, the current view will include only objects specified by the named view.

F4
Shortcut for **Highlighted Objects** on the **View** menu. Crops the view to include only currently selected objects and those objects which are either directly connected to them or which are on a fiber link where the selected objects are involved. The settings of the ***Change View Attributes*** dialog are considered.

SHIFT+F4
Shortcut for **Connected Objects** on the **View** menu. Crops the view to include only the selected object plus the objects logically connected to this object without considering the settings in the ***Change View Attributes*** dialog.

Using SHIFT+F4, the diagram normally contains less objects in the view than using F4.

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** pull down choice. This pull down choice 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 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 online 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 online 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 online help at any time in HCM by using the **Help** menu:

- The **Index** pull down choice displays the index of help topics.
- The **Keyboard** pull down choice displays the Keyboard Shortcuts topic.
- The **About...** pull down choice 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.

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 configuration. 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.

Using the **Customize Field Names** dialog, you can apply meaningful names to the fields as shown in Figure 5 on page 30. You reach this dialog from the menu bar: ***Options → Field Names...***.

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

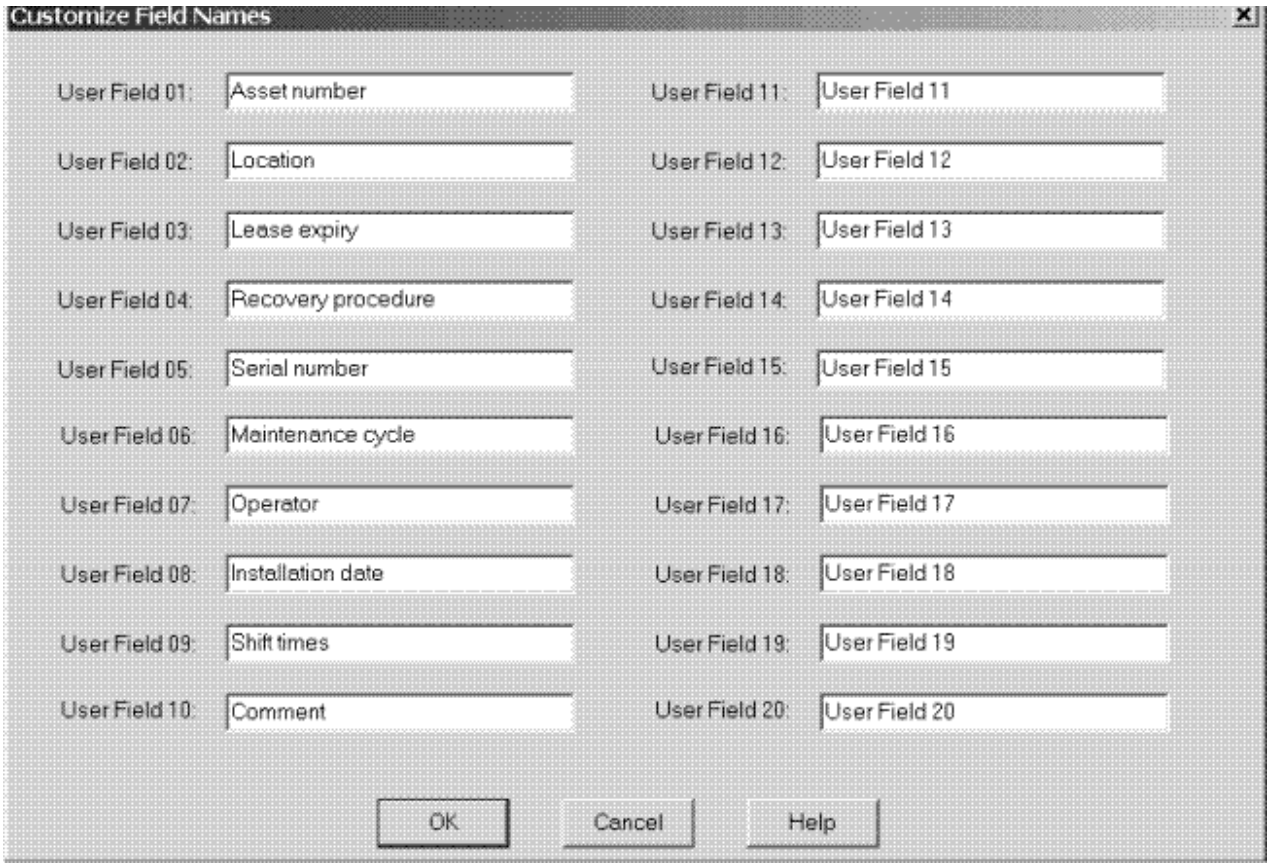


Figure 5. Customize Field Names

You can add or modify your customized user data when editing an object, for example, a processor, by clicking on the **Info...** button on the related **Edit *{object}*** dialogs for these objects. This opens the **User Data** dialog which shows the (customized) user field names with their possibly existing values.

Figure 6 on page 31 shows how you can enter or update your values into customized user fields. Note that only the first ten fields have been customized in this example.

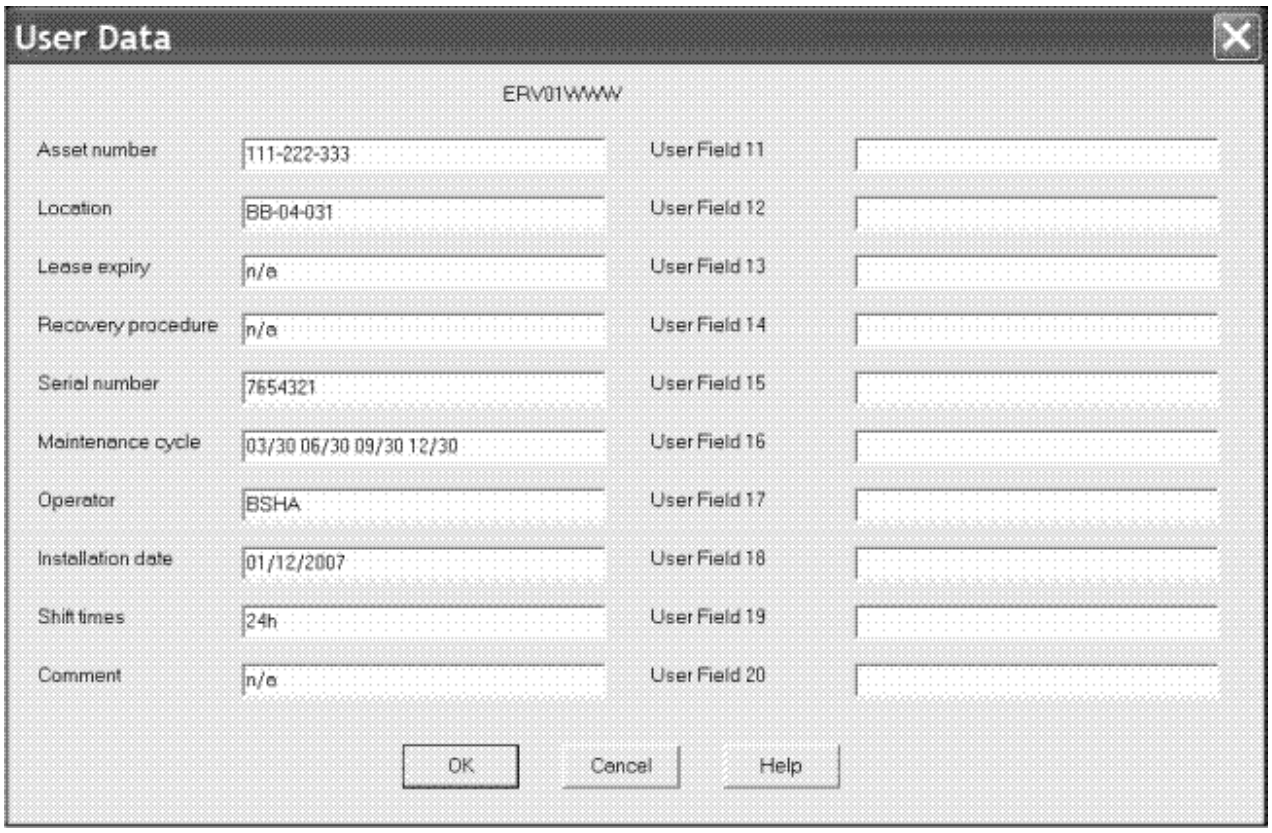


Figure 6. Adding User data into customized user fields

Messages

HCM displays HCD messages as well as HCM messages.

HCD messages

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

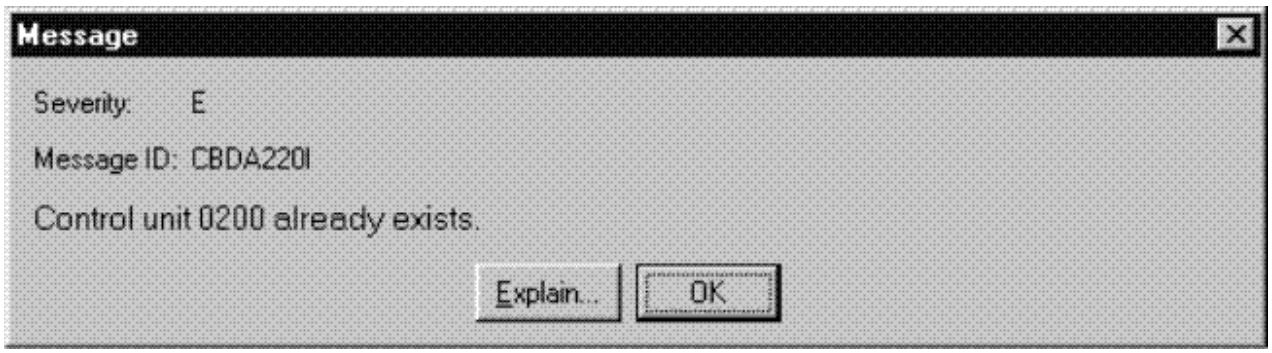


Figure 7. HCD Error Message

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

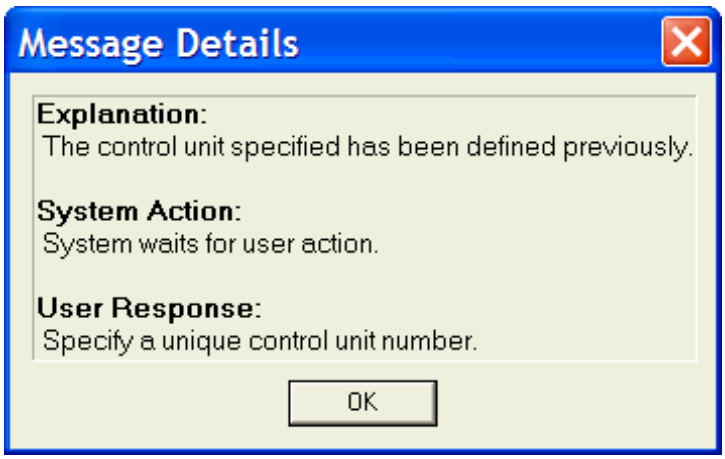


Figure 8. Message Details dialog

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

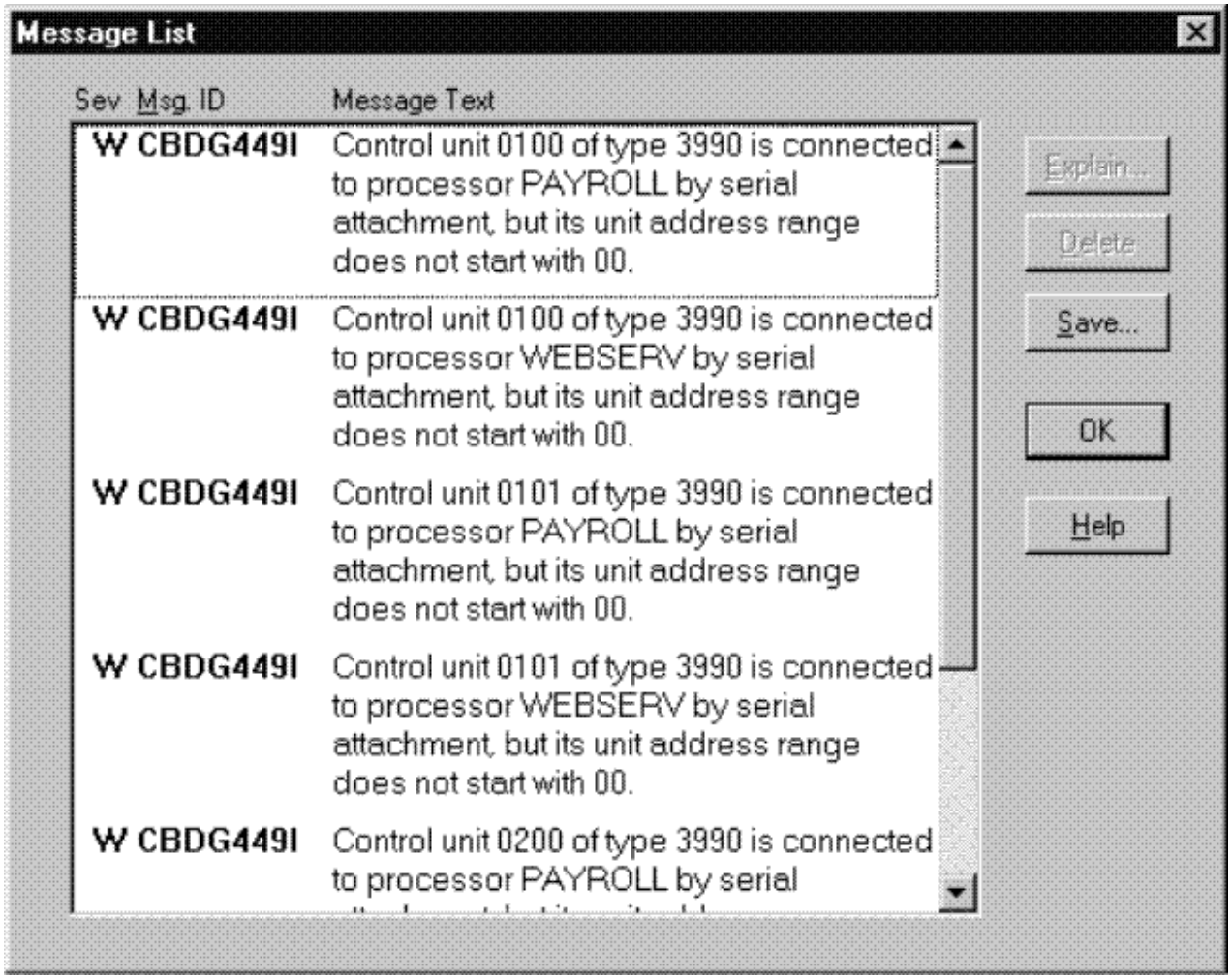


Figure 9. Message List Dialog

In the **Message List** dialog click on the **Save...** push button to save the messages into a file.

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 Chapter 17, “HCM messages,” on page 395.

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 Chapter 17, “HCM messages,” on page 395.



Figure 10. 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 need not worry about how the diagram looks when you create, edit, or delete objects, or when you filter your view. HCM automatically redraws the diagram.

Note:

1. If you position a cursor over an object in the diagram, HCM displays information about this object in the upper left corner of the diagram window beneath the menu bar (fly-over annotations).
2. Information about the current or last selected object is displayed in the lower left corner of the window.
3. In the lower right corner, in connected mode, you see information about the system environment: the host name, the port number, and the sysplex name where the host is a member.

Filtering your view

You can tailor your view of the configuration diagram to focus on the objects of interest only. This results in a more readable diagram. 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.

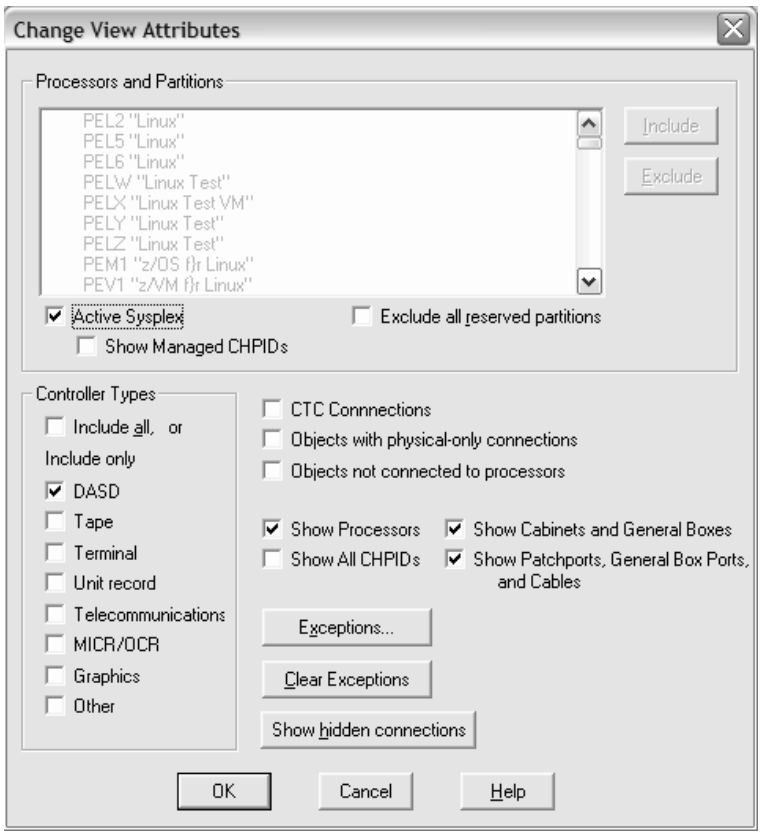


Figure 11. Change View Attributes Dialog

Use this dialog to include and exclude the types of objects you want to see. You can specify which processors, partitions, and controllers should be displayed to reduce the number of objects in the current view. Use the **Include only** list in the ***Controller Types*** section to select the controller device types for just those controllers that you want to see in the view. Press **OK** after you specified all your desired view attributes.

You can use the **Save Named View** pull down choice from the **View** menu to save the current view attributes to a named view. Use the **Restore Named View** pull down choice to select a saved named view to be shown as active configuration diagram.

Note that you may also set *exceptions* to the selected controller types by pressing the **Exceptions ...** button to invoke the **Include/Exclude Controllers** dialog. This dialog lets you add or remove individual controllers from the view regardless of whether or not you included controllers of this device type in the **Include only** list. You can also display objects connected to processors but not to controllers, or objects not connected to processors.

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

By clicking the **Active Sysplex** check box, 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 checkbox is selected, the **Show Managed CHPIDs** checkbox becomes visible 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.

Clicking the ***Exclude all reserved partitions*** checkbox excludes all these partitions from the view in one step, so that you need not select all individual reserved partitions and press the ***Exclude*** button.

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

You can use further pull down choices in the **View** menu to tailor your view:

Highlighted Objects - F4
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 objects and any objects to which they connect. You can print customized cropped views as part of a report. Shortcut: press the F4 key. This applies the settings made in the ***Change View Attributes*** dialog.

The **Highlighted Objects** pull down choice works as a shortcut for setting *exceptions*. For example, if you have the DASD class selected for controllers in the **Change View Attributes** 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 press F12.

Connected Objects - SHIFT+F4
Crops the view to display only the selected objects plus the objects logically connected to this object without applying the settings in the ***Change View Attributes*** dialog. This usually results in diagrams which contain even fewer objects than pressing F4.

If you restore a named view that has been created with SHIFT+F4 , you will see the currently connected objects, even if the connections changed after the view was saved. You can return to the uncropped diagram at any time by pressing F12.

You can also invoke no diagram mode by checking **No Diagram Mode** under the **Options** menu. In the no diagram mode, the diagram is not shown in the current view. In addition, several menus are disabled as there is no diagram to operate, which will result in better performance.

Notes:

- The **Undo** menu is disabled in no diagram mode.
- Toggling the **No Diagram Mode** off enables the **Undo** menu.

Connections in the diagram

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

Table 6. Configuration diagram connections

Line type	**Meaning**
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.

Table 6. Configuration diagram connections (continued)	
Line type	**Meaning**
Stub	Hidden connection. A stub can occur at certain interface objects. It indicates that a connection from or to the interface object is currently hidden in the diagram, either because you decided to hide the connection or because the connected object is not included in the view.

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 38.

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**.

Hiding connections

Using filtering and cropping, you can hide and show certain selected objects. Also, you can explicitly hide visible connections and show hidden ones between certain interface objects. For this purpose, the context menus of:

- CHPIDs
- switch ports
- controller channel interfaces

contain the choices ***Hide Connection*** or ***Show Connection***. The ***Hide Connection*** choice hides the complete connection between two object interfaces, even if it consists of multiple physical cables. The ***Show Connection*** choice makes hidden connections visible again in the configuration diagram.

You can also hide and show multiple selected connections in one step: select the desired interfaces in the diagram and then invoke the appropriate action from the **View** menu bar choice (**View --> Hide selected connections** or **View --> Show selected connections**).

Hidden connections are usually indicated by a "stub" in the diagram as placeholder for the whole connection.

You also can redisplay all currently hidden connections via push button **Show hidden connections** in the **Change View Attributes** dialog (see Figure 11 on page 34).

Saving views

After tailoring a view according to your needs using the filter techniques described in “Filtering your view” on page 33, you can assign a name to this view and save it for later reuse. Thus you can save a set of named views that show different aspects of a configuration.

Choosing **Save Named View...** from the **View** menu, you can save such views that you might want to display again. Whenever 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 processor and partition objects as they are currently drawn on the diagram. When you restore a view, it does not show processors or partitions that you added after saving that view. Subsequently added controllers are displayed depending on the currently active view attributes (see “Filtering your view” on page 33).

The following pull down choices 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.

Vertical view of XMP processors

The default view of a processor in the configuration diagram is the logical horizontal view, where all of a processors's channel subsystems are displayed side by side. If you want to have a more physically oriented view of a processor's spanned channels instead, select the vertical processor view. This view displays the channel subsystems on top of each other and each PCHID appears only once.

You can select your preferred processor view in the **Preferences** dialog that you reach from the ***View ---> Preferences...*** menu bar choice (see "Setting preferences" on page 41).

Figure 12 on page 37 shows the horizontal (above) and vertical view (below) of the same processor.

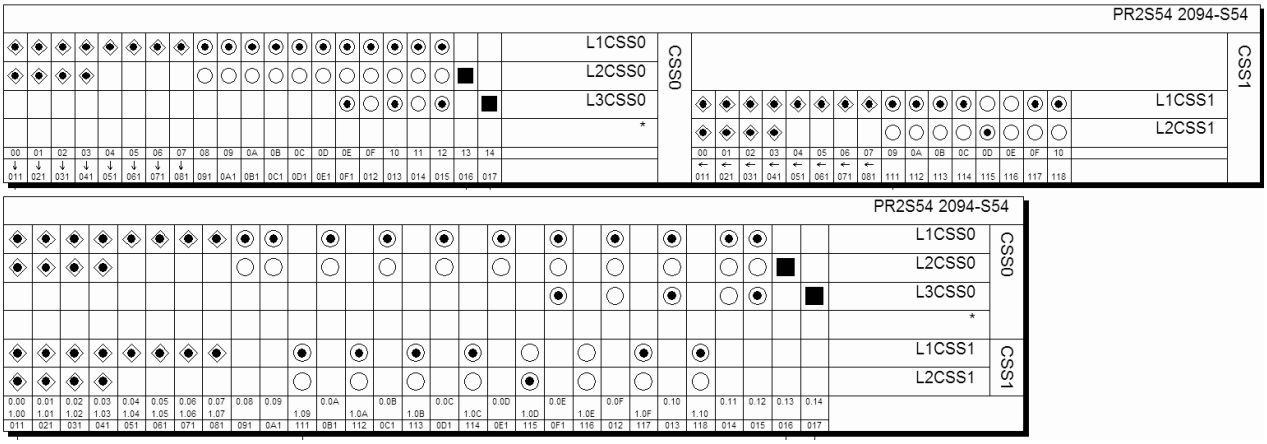


Figure 12. Comparison of horizontal and vertical processor view

Note:

1. In the vertical processor view, you can see only one (physical) instance of a spanned CHPID. Nevertheless, in such a case, you can get information about all logical instances from such a CHPID's context menu (see Figure 13 on page 37).



Figure 13. Context menu of a spanned CHPID in a vertical processor view

2. Some dialogs concerning spanned CHPIDs make a difference whether you invoked them for a logical instance of a CHPID from the horizontal view or for the single physical instance from the vertical view. Examples for such dialogs are the **Edit CHPID** dialog or the **Assign Cable** dialog.While a certain logical instance in the last-mentioned dialog is referred to as, for example, `PR3S54.0.94`, the underlying physical instance selected from the horizontal view is designated as `PR3S54.[0/1/2].94` if it is spanned over CSSs 0, 1, and 2.

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:

Diagram object	Meaning
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)

Diagram object	Meaning
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)

Diagram object	Meaning
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.

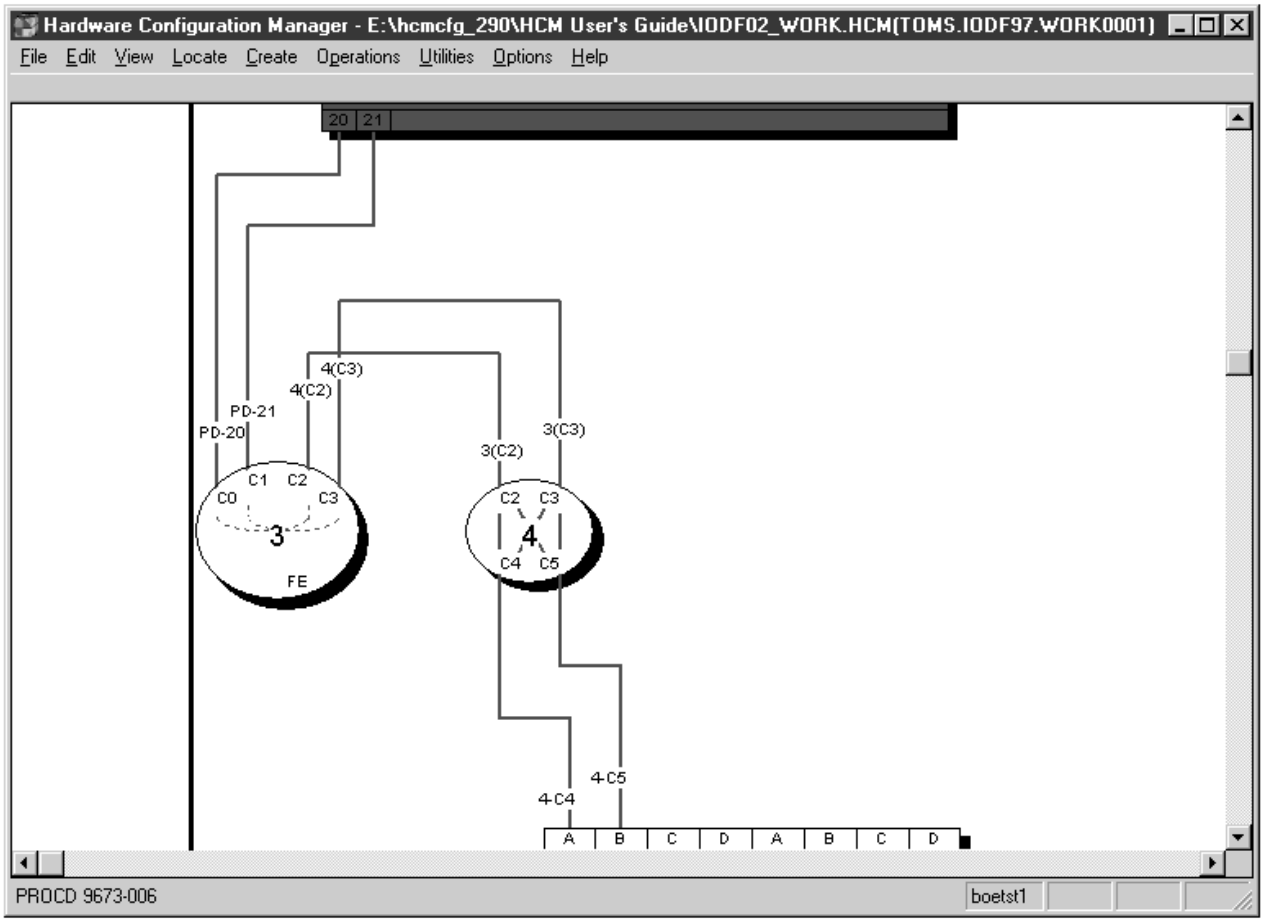


Figure 14. Highlighting an Object, without Visualized Configuration

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.

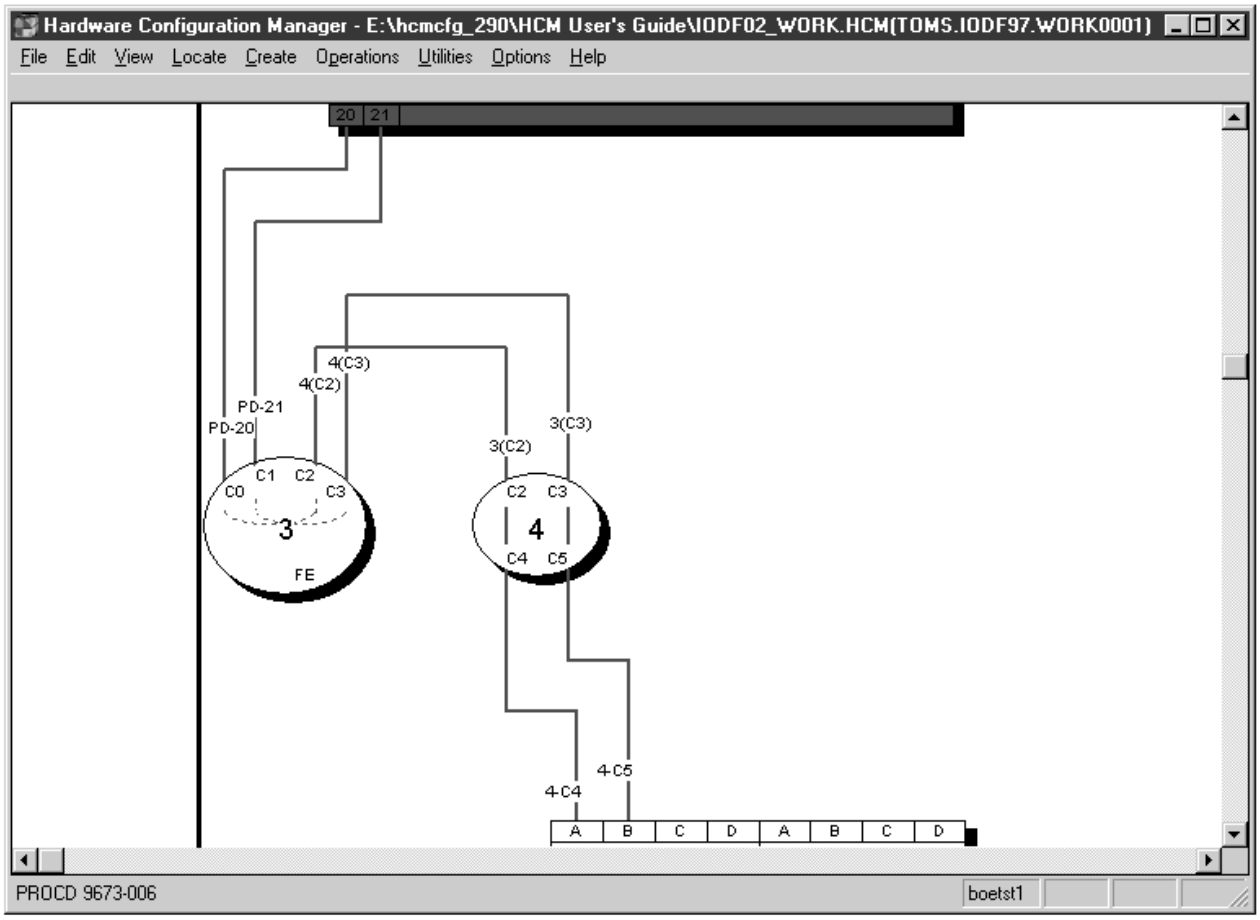


Figure 15. Highlighting an Object, with Visualized Configuration

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 pull down choices 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 28. 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 28 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** pull down choice from the **View** menu.

Setting preferences

Using the **Preferences** dialog from the ***View ---> Preferences...*** menu bar choice, you can select alternative settings how to display the configuration, which will be saved for future HCM sessions:

Processor view
With this setting you can modify the view of XMP processors: Select the horizontal processor view to display the channel subsystems side by side. Select the vertical processor view to display channel subsystems on top of each other. In the vertical view, you achieve a physically oriented view of spanned CHPIDs and their PCHID value appears only once in the diagram.

Channel and Device Adapter Grouping
For controllers that have channel or device adapters, use this option to choose whether these 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 **Grouped by Adapter** will order the adapters: 0A,1A,0B,1B,0C,1C,0D,1D. Choosing **Grouped by Segment** will order the adapters: 0A,0B,0C,0D,1A,1B,1C,1D.

Converter shape
Display all converters in the diagram as either elongated hexagons or pentagons.

Connector lines
With this setting, you can select whether connections between objects should always be drawn with rectangular (horizontal and vertical) lines or whether diagonal lines should be used where possible.

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 pull down choices 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.

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 about the resynchronization process, see “Resynchronizing configuration files and IODFs” on page 53.

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 33). 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 11, “Reports and diagrams,” on page 263 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 120) 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

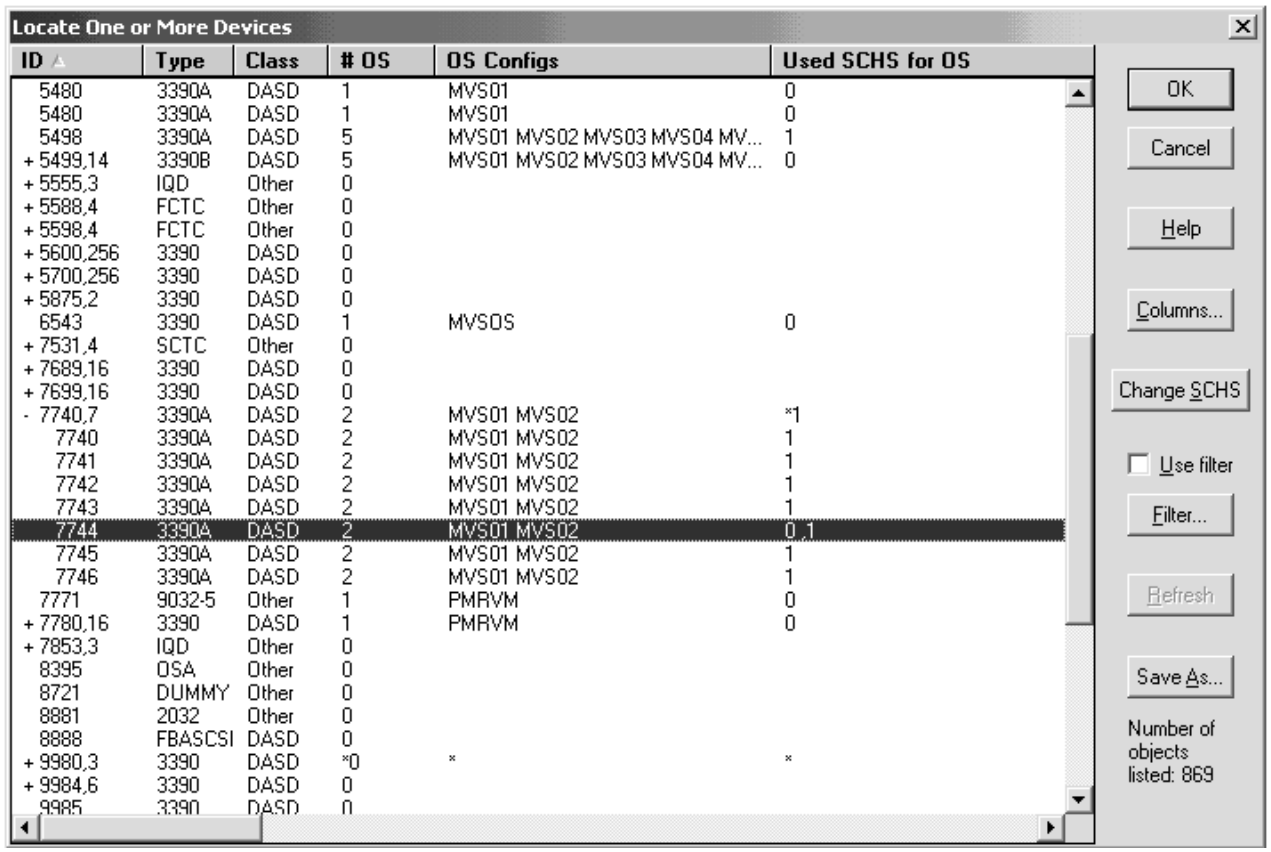


Figure 16. Selection List for Locating Objects, in this Example: Locating Devices

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 43.

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

With the button **Change SCHS**, which appears only on the ***Edit a Device*** and the ***Locate One or More Devices*** dialog, you can change the placement of (multiple) eligible devices in a subchannel set. This is further described in “Changing the subchannel set for devices” on page 149.

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.

Filtering objects

Use the **Filter...** button in the ***Locate*** and ***Edit*** dialogs to reduce the list of objects displayed in the resulting window. HCM opens a dialog in which you can specify filter criteria for the objects you want to locate or edit. These criteria are stored across HCM sessions.

Figure 17 on page 45 shows a filter dialog for CHPIDs:

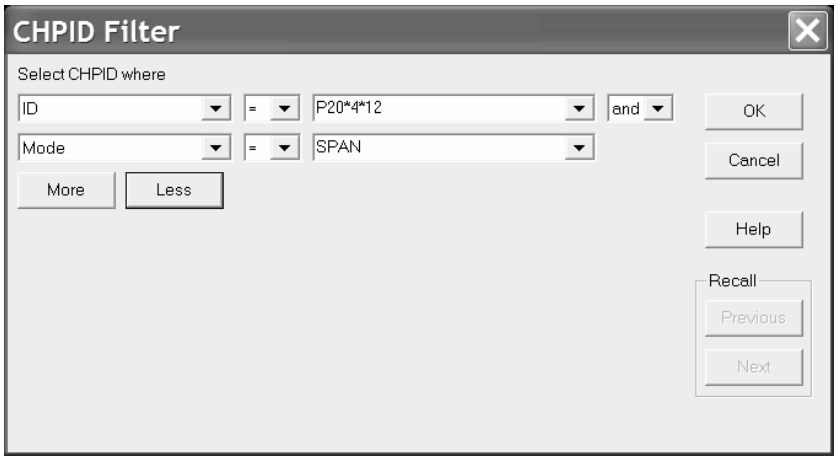


Figure 17. Filter dialog

This dialog provides up to eight lines of entry fields where you specify the filter criteria. For example, using wildcards, you can filter for all CHPIDs whose IDs obey to a certain name schema, as shown in Figure 17 on page 45 or you can look for CHPIDs in a specific range. To specify your criteria, in the third entry field of each row, you can use the wildcard character '*' multiple times, at any position, with each wildcard representing an arbitrary number of characters.

Note: After editing an object, it might not meet your filter criteria any longer. To update the list, press the **Refresh** button.

Tip:

You can also use an object list with its filter and arrangement capabilities to analyze the content of your configuration and to create a user-defined report using the **Save As...** button.

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 99.

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 43.

For more details, see the section for the specific **Edit...** dialog in Chapter 6, “Modifying objects in your configuration,” on page 99.

The **Interface Selection...** pull down choice 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 99 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 42 for more details.

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

Note 1:

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.

Note 2:

Besides the possibility to delete multiple objects via selection on the configuration diagram, users can now select multiple objects on some **Edit *<object>*** dialogs and delete them in one step by pressing the **Delete...** button on these dialogs.

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. Press the **Order...** button in the appropriate dialog. For processors, select the **Order Processors...** command from the **Edit** menu. The standard order dialog appears 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.

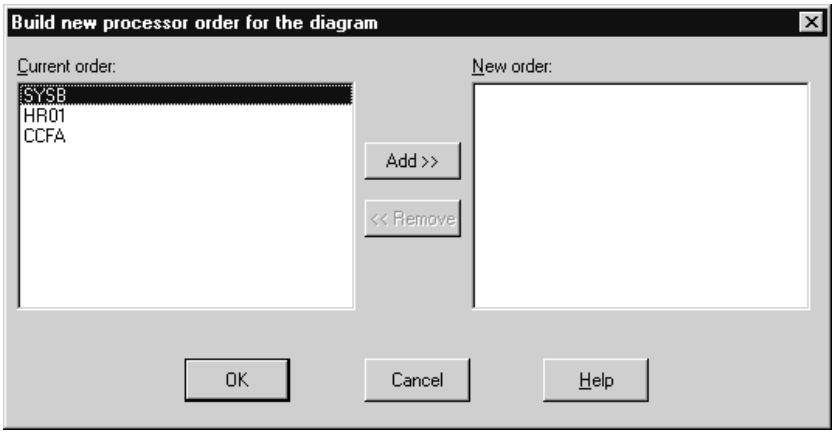


Figure 18. Build New Processor Order for the Diagram

Chapter 4. Building your configuration

HCD uses the IODF to store the logical configuration data defined for the operating system and the channel subsystem. HCM, running on a PC, acts as a "graphical front end" to HCD: loading the IODF into HCM allows you to build an associated, interactive configuration diagram in a configuration 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 configuration 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 from the scratch or by populating a new configuration file with data from an existing, prepared IODF. You can open and/or resynchronize an existing configuration by opening the configuration file with its associated IODF or against a different IODF. Copy the configuration 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 99 or Chapter 7, "Connecting objects in your configuration," on page 175). When you are finished, close your configuration. This saves your changes in the config file and associated IODF.

Read the following topics to learn how HCD and HCM keep their configuration data consistent and how you can create and work with HCM configuration files:

- "HCM and HCD" on page 49
- "Physical and logical objects in HCM" on page 54
- "Introduction to the master configuration file concept" on page 54
- "Creating a new configuration from an IODF" on page 56
- "Creating a new configuration from scratch" on page 66
- "Opening an existing configuration" on page 67
- "Working with your IODFs and configuration files" on page 71

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 configuration file in sync with the IODF.

The *IODF* contains only logical definitions; the *configuration file* contains a subset of the logical data from the IODF, plus physical information added using HCM. Together the IODF and the configuration 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 does not differ from an IODF created in HCD; no HCM data is stored in the IODF.
- HCM owns the data in the configuration file; all validation of the physical data and error messages relating to that data come from HCM.

Each configuration file is associated with a single IODF; think of the configuration file as an extension to the IODF. When you open the configuration file, its associated IODF is automatically opened. You can

© Copyright IBM Corp. 1995, 2021

open the configuration file against a different IODF, for example, if the associated IODF was damaged or renamed in HCD.

You can have multiple configuration files associated with a single IODF. However, this is not recommended. Every time the IODF is updated through one of its configuration 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 52).

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 a configuration" on page 76).

You may wish to backup the IODF and configuration file before making changes. Use the **Copy Configuration Files...** pull-down choice of 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 a configuration" on page 76 for more details.

Relation of configuration files and IODFs

With HCM you can create *work configuration files* and *production configuration files*. These configuration files are associated with the IODF data sets, forming a work or production pair. Figure 19 on page 50 illustrates the relationship.

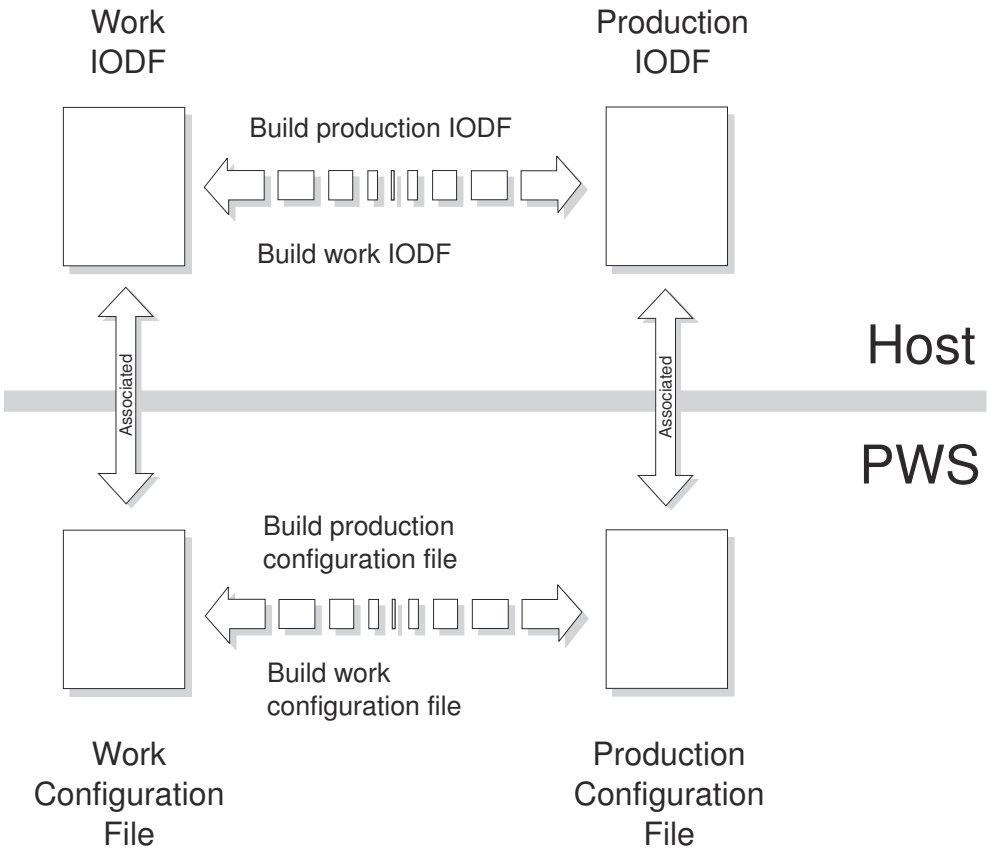


Figure 19. Configuration Files and IODFs

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/*.hcmz) file or a production configuration (*.hcr/*.hcrz) file on the PC is created.

Note: If you choose the option to compress your configuration files, they get the extensions *.hcmz or *.hcrz (see "Compressing configuration files" on page 78).

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 about all dialogs, diagrams, and reports in the stand-alone mode. For more information refer to Chapter 5, "Stand-alone mode," on page 95. 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 online 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 72 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 online 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 data
- Logical data of control unit - processor definitions
- Logical data of device - processor definitions

HCM configuration files

The term *configuration file* is used for both *work* configuration files (*.hcm/*.hcmz) and *production* configuration files (*.hcr/*.hcrz).

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" on page 52 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 configuration file. HCM 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 configuration file, *provided none of the users is updating*. HCM initially opens the IODF and configuration 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 configuration 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 z/OS UNIX for your workstation. You then can access the file system of z/OS UNIX as if it were a local drive or a LAN drive. Storing an HCM configuration file on the file system of z/OS UNIX 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 z/OS UNIX as well as a home directory in z/OS UNIX. Hence, you could also save HCM configuration files in your z/OS UNIX directory (or any other location in the z/OS UNIX file system where you have access).

The IODF load / resync process

In addition to the physical data which resides only in the configuration 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 configuration file, HCM

does not have to re-read it from the host each time the configuration 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 configuration file is created from an IODF (*IODF Load*). See "Loading the IODF" on page 59 for more details.
2. When the IODF on the host is modified outside of HCM by using, for example, the HCD ISPF interface. HCM detects that the IODF on the host is out of sync with the local copy, and re-reads the local IODF data when the configuration file is next opened (*IODF Resync*).
3. Updating only one configuration 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 configuration 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 is 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 configuration file. If you specify an already existing configuration file name, all currently stored information in the existing configuration file is completely overwritten. It is lost.	See "Loading the IODF" on page 59
Resynchronizing IODF and configuration file	The local copy of all logical configuration data in the configuration file is overwritten with the data of the IODF. The physical data is generally kept.	See "Resynchronizing configuration files and IODFs" on page 53

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 is overwritten with the data of the IODF. Generally, the physical data is kept.

The resynchronization operation is indicated by the following dialog:

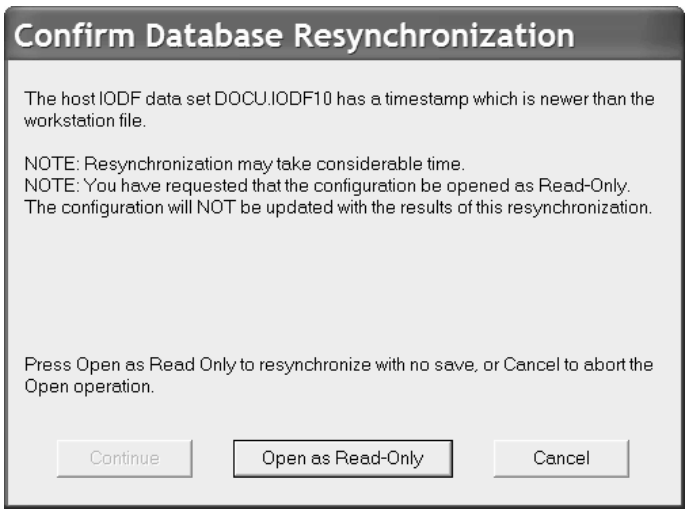


Figure 20. Confirm Database Resynchronization dialog

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 configuration file or save it, resolve these conflicts. See "Receiving visual warnings" on page 42 for information about 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.

Introduction to the master configuration file concept

The HCM master configuration file (MCF) is an HCM configuration stored on the host. It provides a central shared repository, allowing several HCD/HCM users to work on a single configuration cooperatively and safely. This does not mean concurrent access to the configuration: the MCF enforces sequential access, one user (in connected mode) at a time. Physical changes committed to the MCF by one HCM user are automatically made available to the next user.

The master configuration file also helps to resolve situations when one user works on a configuration in connected mode, performing physical and logical updates and another user works on a local copy of the same configuration in stand-alone mode, doing physical-only changes. If this user accesses the configuration plus the related IODF later, a physical mismatch resolution dialog is invoked. This ensures that consistency is maintained for physical configuration changes among multiple copies of a configuration.

Note: The MCF feature is new with z/OS 1.9. Its exploitation is optional. A detailed description is provided in "Using the master configuration file (MCF)" on page 79. Also, several usage scenarios are provided in "How to work with MCF-enabled configurations: Scenarios" on page 82.

Without using this feature, processing of HCM configurations continues to work as in previous releases.

Physical and logical objects in HCM

The IODF contains information about *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 usually have corresponding physical routes. However, situations may occur where the logical definitions and physical routes do not match. HCM highlights these cases in the configuration diagram; see "Receiving visual warnings" on page 42 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 logical processors in the IODF which belong to the same physical processor 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. If you have more than one device with the same device number, HCM uses the serial number to determine whether the devices are physically identical. Usualla, it is sufficient to assign serial numbers to just the first device in each string, as devices with the same serial number are placed in the same string. 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 57 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 57).

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 buses, depending upon manufacturer and model.)

2. For controllers, which have a hierarchical internal architecture, such as the interfaces within channel adapters contained in buses (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 is 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 finds 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 is used. If two or more are found, users are 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) assigns 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 (for example, 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 15, “Physical description files (PDF),” on page 379 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 configuration file. See Chapter 14, “Importing and exporting data,” on page 333 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 66.

Essentially, you create a new configuration by loading data from the desired IODF on the host into a configuration file which is created on the PC. 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 need to rectify this by preparing the IODF before loading.

HCM can recognize CTC connections, if all control units 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.

If the used IODF is MCF-enabled ...

... HCM informs you with a message, that it is going to create and open your new configuration file as a copy of the downloaded MCF. In such a case, your new configuration may already contain physical definitions, like cables or general boxes. Read more detailed information:

- "Loading the IODF" on page 59
- "Loading an IODF with associated MCF" on page 82

If the used IODF is MUA-enabled ...

... be aware of how you can work with and how HCM handles such configurations. Refer to "Working with multi-user access enabled IODFs" on page 62.

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 "Creating work and production IODFs without the associated configuration files" on page 74). 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 is incomplete and you get logical-physical inconsistencies (see "Receiving visual warnings" on page 42). 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 54) 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 configuration file, update the IODF using HCM, discard the configuration file, and then reload the IODF.
2. Use HCD to update the IODF before loading an HCM configuration 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 is described 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 59. Any physical information that requires updating is 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.

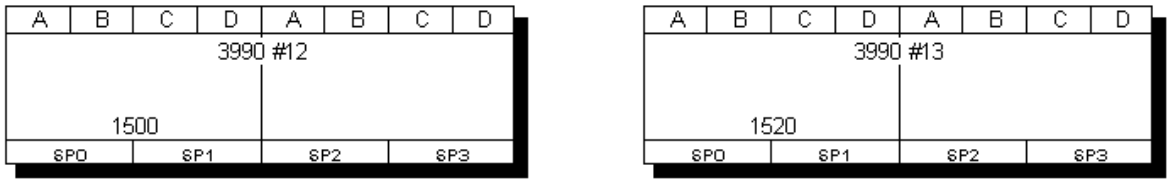


Figure 21. After loading the IODF in which control units have no or different serial numbers.

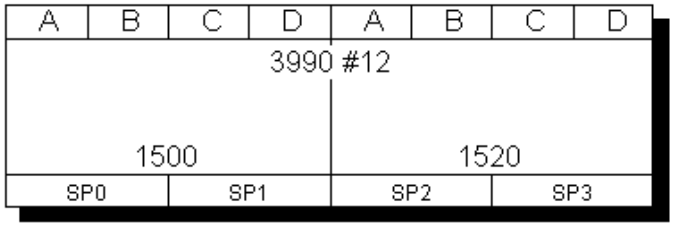


Figure 22. After loading the IODF in which control units have the same serial numbers.

Do not be too concerned if the physical configuration diagram is inaccurate at this point. Many of these problems are 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 54 for an explanation of how HCM relies on serial numbers. Pay particular attention to I/O devices which have more than one device number (for example, 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 to 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 131 for controllers
- "Choose physical description" on page 142 for strings
- Chapter 15, "Physical description files (PDF)," on page 379

See Chapter 15, "Physical description files (PDF)," on page 379 for more details. If you have "brand X" equipment, ensure that you have read and understood the "Graphical representation of physical objects" on page 55. 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 configuration file. This implicitly saves the IODF. Then discard the configuration file by closing and deleting it. Reload the updated IODF into a new configuration 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 99).

Loading the IODF

This step loads the IODF data set into a configuration file. If the configuration file already exists, HCM requests you to confirm that it is OK to completely overlay this 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 23 on page 59). 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.

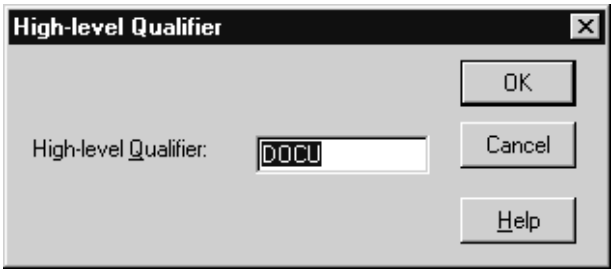


Figure 23. High-level Qualifier dialog

The contents of the list of existing host IODFs displayed in the **IODFs** dialog depends on the qualifier specified (see also “Specifying a high-level qualifier for z/OS” on page 63. The currently used high-level qualifier is always displayed below the **Qualifier...** button on the **IODFs** dialog. In the example in Figure 24 on page 60, the high-level qualifier specified is "DOCU".

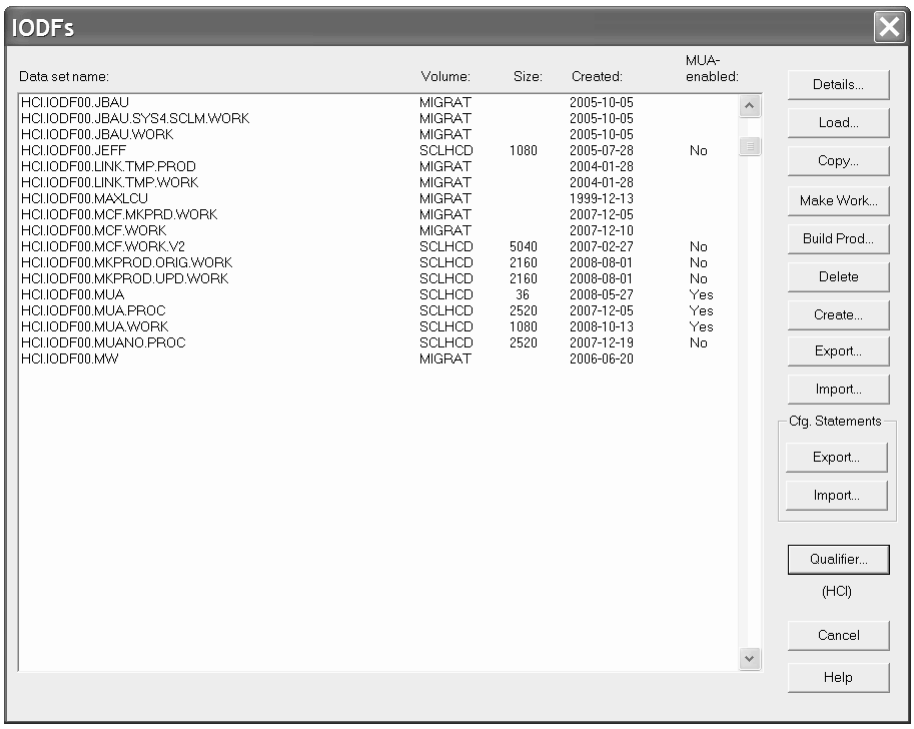


Figure 24. IODFs dialog

Note: For migrated IODFs (identified by MIGRAT in column ***Volume***), HCM cannot determine the MUA status and the size, and therefore columns ***MUA-enabled*** and ***Size*** remain blank.

Select an IODF and click on the **Load...** button to request the loading of the selected IODF into a configuration file.

Once the **Load...** function starts, HCM displays the **Choose Local File** dialog (Figure 25 on page 61). You are then taken through steps to select or create a configuration file to receive the IODF information. In other words, the **Load...** function creates a new configuration based on the contents of the IODF. The configuration file is automatically associated with the selected work IODF.

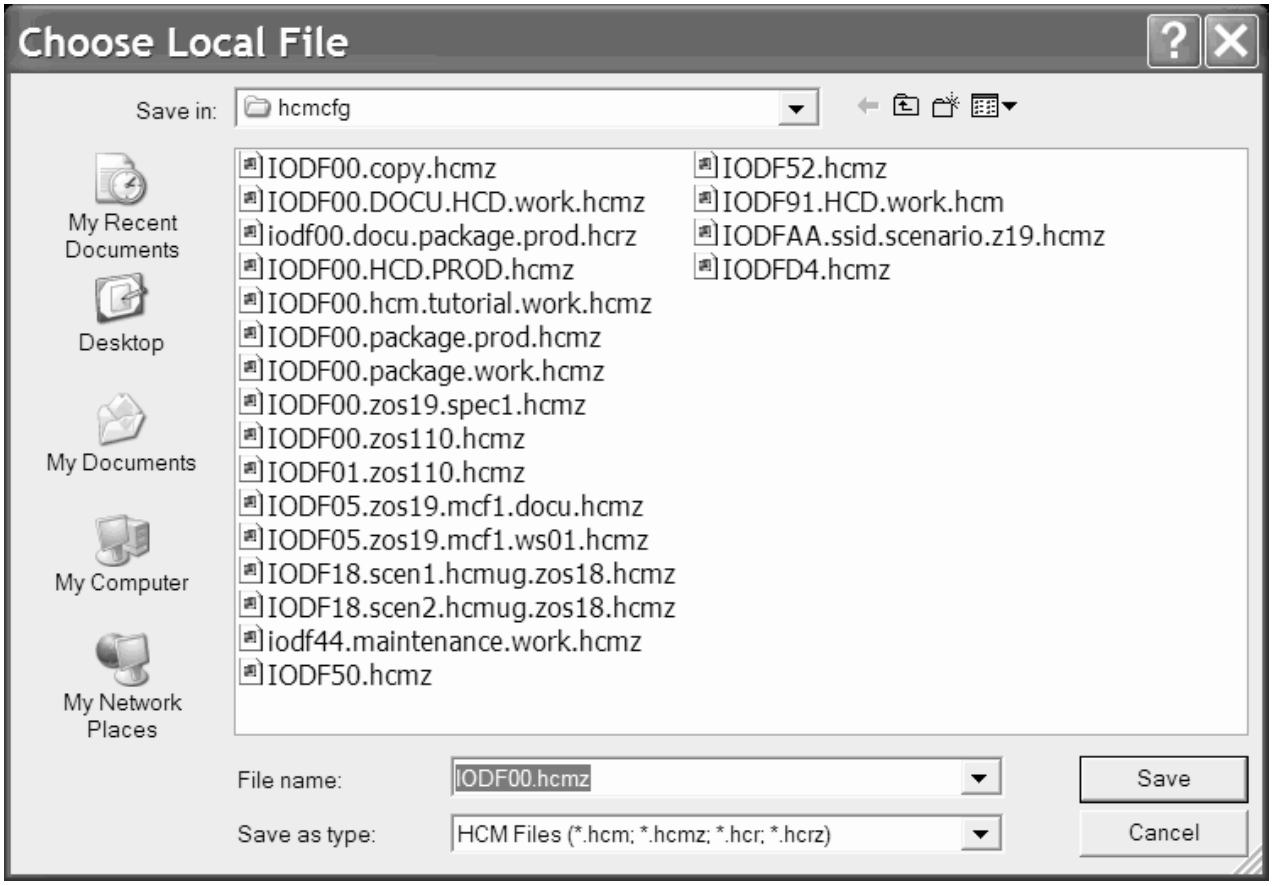


Figure 25. Choose Local File dialog

The **Choose Local File** dialog suggests a default directory where to store your configuration. You can change to a different directory as storing location. Your selection is saved across HCM sessions.

Tip:

If you specify an already existing configuration 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 configuration 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.

Loading a work IODF with associated MCF

If the IODF that you want to load into a configuration has an associated MCF, HCM informs you with a message, that it is going to create and open your new configuration file as a copy of the downloaded MCF. In such a case, your new configuration may already contain physical definitions, like cables or general boxes.

Press the ***OK*** button on this message to start the process of:

- downloading the MCF into the specified configuration file
- opening the configuration as a diagram
- and implicitly associating the configuration to the IODF (same as without MCF).

Also, HCM automatically MCF-enables your new configuration. In contrast to loading an IODF without associated MCF, you also see all physical definitions previously applied to the configuration.

Before loading the IODF into a (new) configuration, you may want to find out whether this IODF has an associated MCF data set or whether it is enabled for multi-user access (MUA). Select the IODF from the

list of existing IODFs in the ***IODFs*** dialog and press the ***Details...*** button to display the ***IODF Details*** panel (Figure 26 on page 62) which indicates the status of the IODF (MCF/MUA status enabled or disabled).

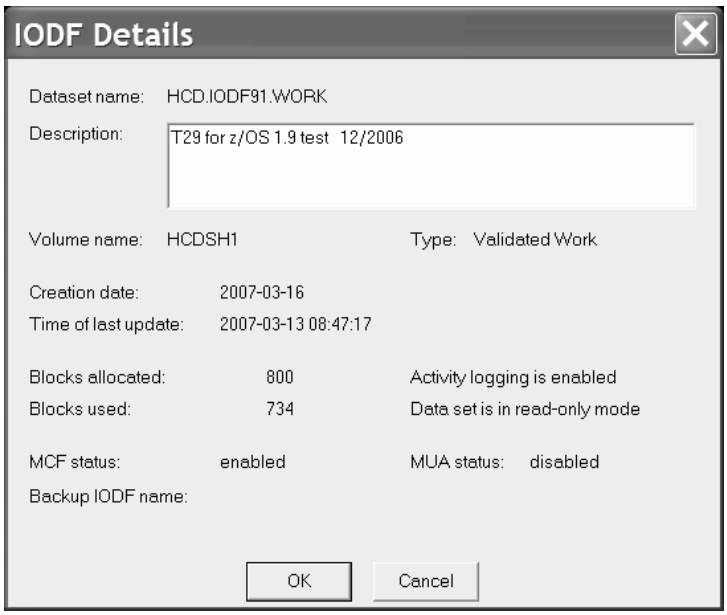


Figure 26. IODF Details

Loading a production IODF with associated MCF

If you are loading a production IODF with an associated MCF, HCM downloads the MCF and opens the master configuration. Since the master configuration associated with a production IODF is always a ***work*** configuration file, you now have an in-sync, up-to-date configuration, but you may not be ready to continue to work with it.

Note:

When loading a production IODF, HCM always creates a work configuration file first. Therefore, hcm(z) is offered as the default file type in the *Choose Local File* dialog. This behavior is not new with the MCF functionality. The created work configuration is immediately usable only for viewing or for physical-only updates.

When loading an MCF-enabled production IODF, HCM offers to additionally create a production configuration file. Creating an associated production configuration file is recommended, but not mandatory (for example, for just viewing an up-to-date state of the master configuration).

Even if you do let HCM create the production configuration, it still opens the downloaded work configuration file first. This is to keep the behavior consistent with older releases of HCM. At this point, you cannot apply logical changes because the associated production IODF cannot be changed. Also, the menu choice ***File ---> Make Work*** is disabled, because HCM enables it only for production configuration files. If you want to just view an up-to-date state of the master configuration (or if you want to apply purely physical changes), the opened work configuration is sufficient for that purpose.

Often however, you now want to apply (physical and logical) changes to a new pair [work IODF, work configuration file]. In order to be able to create such a pair, you first have to close the work configuration and open the production configuration (which you previously allowed HCM to create during loading of the MCF-enabled production IODF). Here, the menu choice ***File ---> Make Work*** is enabled and allows you to create the new work configuration pair.

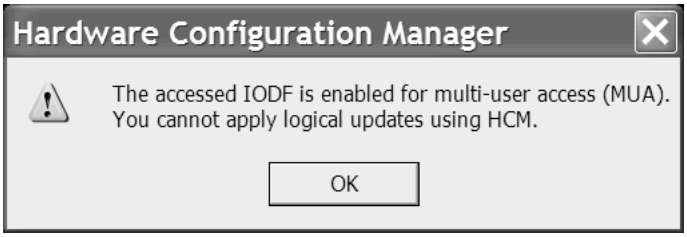
Working with multi-user access enabled IODFs

For an IODF on the host, HCD users can specify a multi-user access (MUA) option, so that multiple users can simultaneously update this IODF using HCD.

HCM provides information about an IODF's MUA status, either in the ***IODFs*** dialog or in the ***IODF Details*** dialog, so that before working with an IODF in HCM, you can check its MUA status (enabled or disabled).

HCM users can work with MUA-enabled IODFs in the following way:

- You can load a multi-user access enabled IODF from HCD into a new HCM configuration file and you can open an existing HCM configuration associated with a MUA-enabled IODF. HCM informs you when the accessed IODF is MUA-enabled:

  After pressing ***OK***, you can apply physical changes and you can use the viewing and reporting features of HCM on this configuration. However, you cannot apply logical updates that would affect the associated IODF.
- When opening a configuration with a multi-user access enabled IODF, you see and keep the current status of the IODF during your complete HCM session. You will not be aware of any changes that other HCD users might apply during that time.

  If you close the configuration and re-open it at a later point in time, HCM will conduct any required resynchronization (see "Resynchronizing configuration files and IODFs" on page 53).
- Operations on ***MCF-enabled*** configurations work in the same way for associated ***MUA-enabled*** or ***MUA-disabled>*** IODFs, with one exception: You cannot change the MCF status of a configuration associated with a MUA-enabled IODF.

For more information about multi-user access on IODFs, refer to the *z/OS HCD User's Guide*.

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 displays the list of IODFs beginning with that qualifier.

z/VM:

For z/VM this dialog is not applicable. There is no filter to reduce the number of listed IODFs in the **IODFs** dialog. Instead, all IODFs are displayed that are available on your accessed minidisks as shown in Figure 27 on page 64.

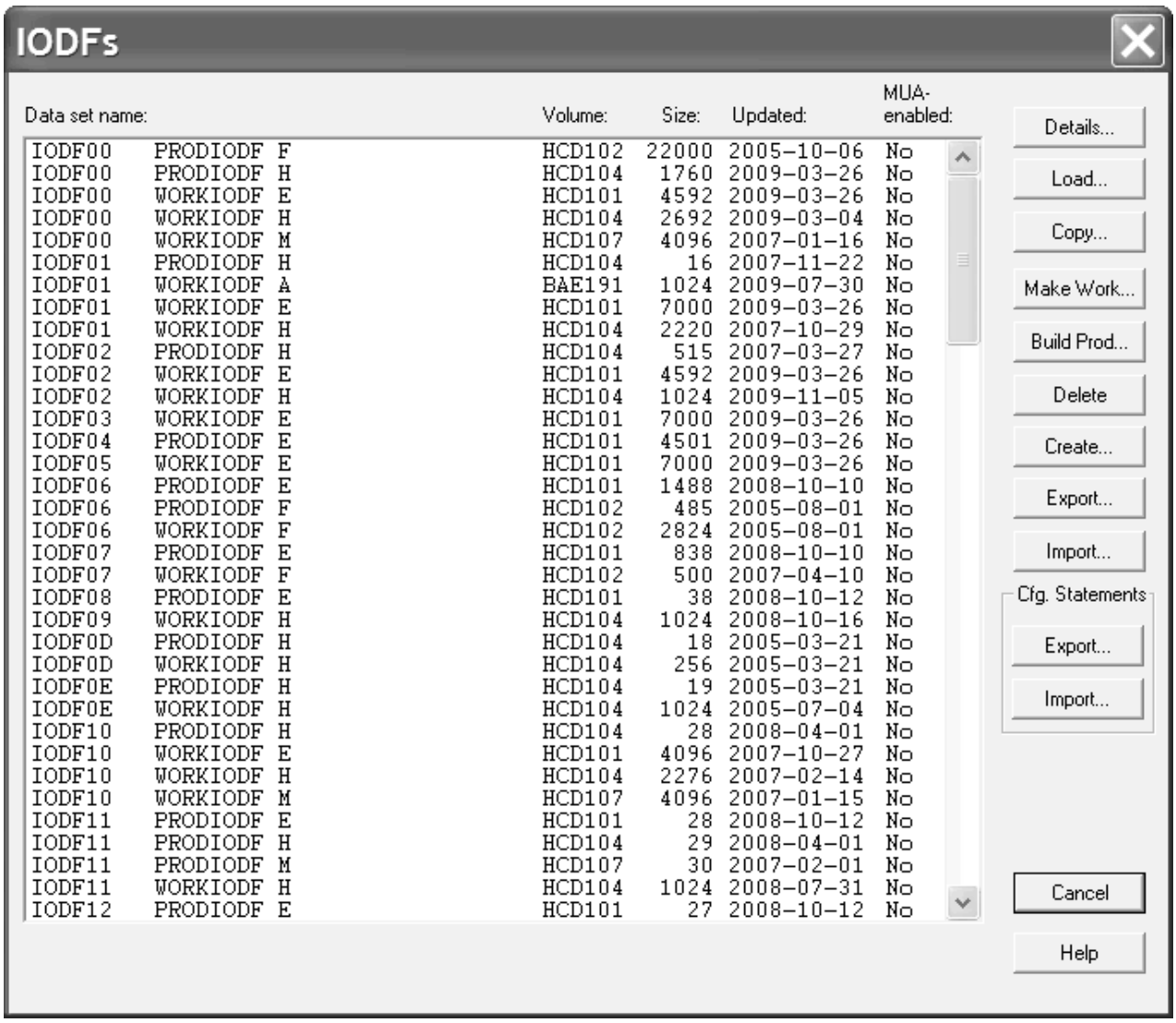


Figure 27. IODFs dialog for z/VM

Choosing alternate processor definitions

If there is more than one processor defined in the IODF, HCM asks 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 automatically appears 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.

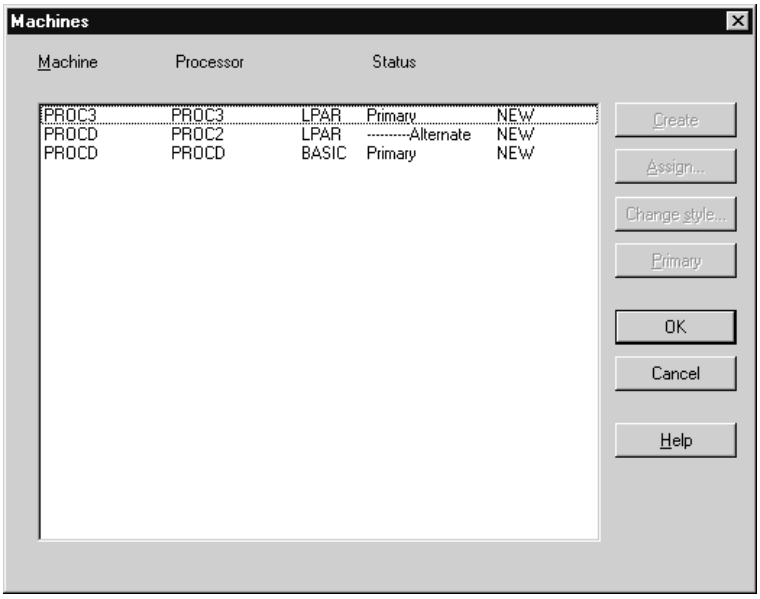


Figure 28. Machines dialog

HCM attempts to assign processors and primary / alternate status intelligently. The **Machines** dialog asks you to confirm or change HCM's assignments; typically, you press OK to confirm. For details on how to change HCM assignments, and how to manage shadow processors in general, see Appendix A, “Alternate processor definitions,” on page 415.

After loading the IODF, the PC configuration file and the IODF are saved automatically.

Inspecting the initial configuration

The result of the IODF load process is the creation of an initial configuration in a configuration 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 reload 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 reload the IODF, or use HCM to delete the controller and control unit(s) and recreate and reconnect 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:

1. After you have made any physical changes, do not load the IODF again, because a load replaces the existing configuration file and all physical data will be lost.
2. If you loaded an IODF with an associated MCF data set, HCM has created your new configuration as a download copy from that MCF, so that it most probably already contains physical definitions.

How to update physical data is described in the following topics:

- Changing PDF assignments for controllers and strings (see “Graphical representation of physical objects” on page 55 and Chapter 15, “Physical description files (PDF),” on page 379).

- Moving control units between controller segments (see “Arranging controller components” on page 134).
- Ordering the interfaces on controllers and strings (see “Arranging controller components” on page 134).
- Ordering daisy-chained controllers and strings in the proper sequence on their cables (see “Connecting objects” on page 175 and “Positioning objects along daisy-chained connections” on page 176).
- Creating and connecting crossbar switches and fiber management cabinets (see “Crossbar switches” on page 162 and “Cabinets” on page 165 and “Editing fiber links” on page 212 and “Connecting crossbar switches” on page 218).
- Importing physical objects, changing physical connectivity, updating descriptive fields in both the PC file and the IODF, resynchronizing the PC file and the IODF, (see Chapter 14, “Importing and exporting data,” on page 333).
- Managing cables (see Chapter 8, “Physical connections,” on page 207).

After you have made physical changes, use the **Copy Configuration Files...** pull-down choice of the **File** menu to make a backup copy of your configuration 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 configuration 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.

After a successful creation, the new IODF is added to the list of IODFs displayed in **IODFs** dialog.

Then, on the same dialog, with the new IODF selected, press the **Load...** button. This offers you a dialog to specify and save an HCM configuration file which is associated to the new HCD IODF. Now you are ready to create configuration objects. More information is in “Loading the IODF” on page 59.

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 allows you to define at least 4000 devices, in general. You cannot change the space for an IODF 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 71).

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 59.

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

WORKCLOG
change log file for automatic activity logging

Examples:

```
IODF01 WORKIODF A
IODF01 PRODIODF A
```

For more information about naming conventions for z/VM HCD refer to *z/VM: I/O Configuration* (SC24-6100).

To populate a new configuration file with data from an existing IODF, see "Creating a new configuration from an IODF" on page 56.

Opening an existing configuration

If you are opening an existing configuration to edit, you can either:

- Open an existing configuration file and associated IODF, or
- Open an existing configuration 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 to ensure they are in sync.

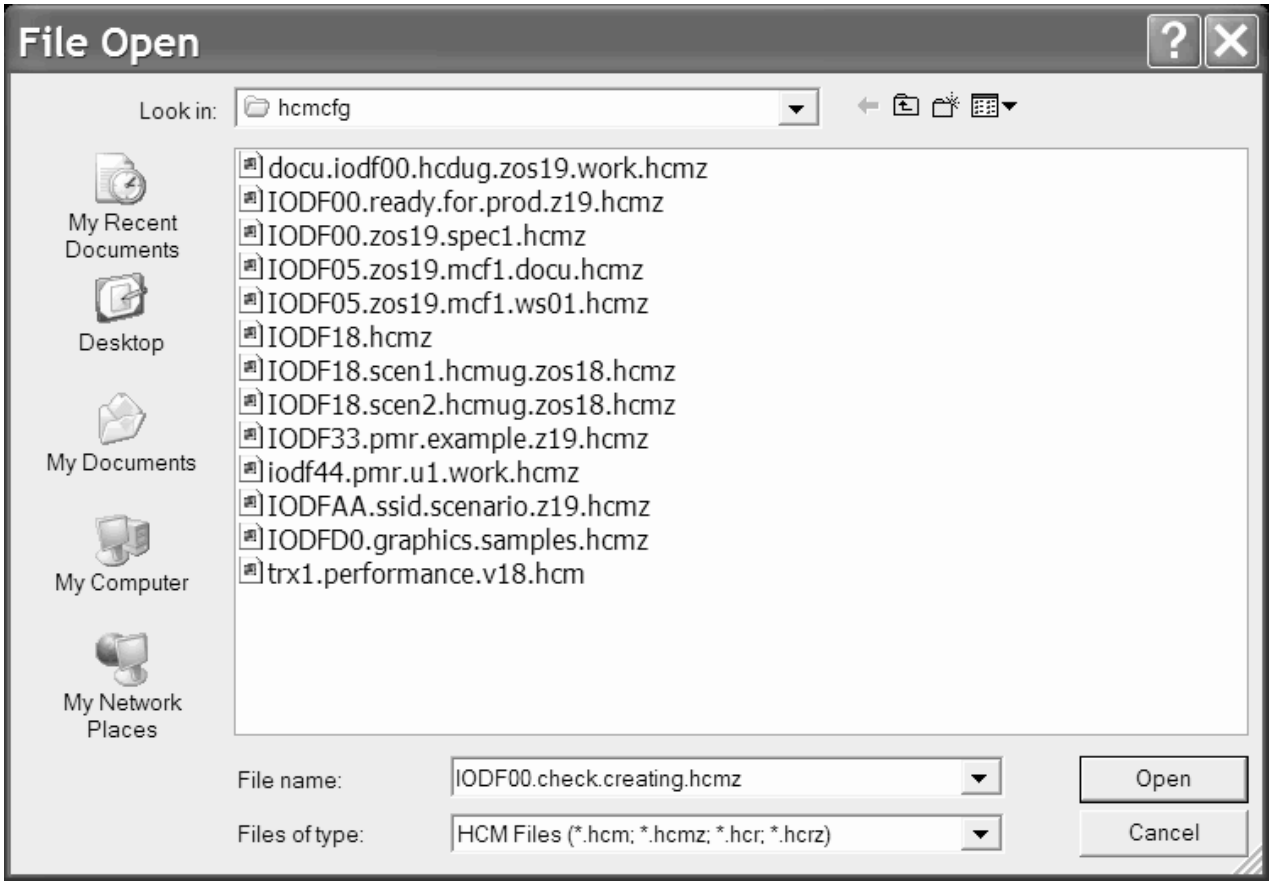


Figure 29. File Open dialog

If the configuration file and IODF are out of sync, HCM asks 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 configuration file, HCM will resynchronize; the configuration file will be altered to match the changed IODF.

If the configuration is MCF-enabled ...

1. ... and HCM finds a more recent MCF than the used existing configuration and **there are no** stand-alone mode changes in this configuration, then the MCF automatically replaces the existing configuration after requesting confirmation.
2. ... and HCM finds a more recent MCF than the used existing configuration and **there are** stand-alone mode changes in this configuration, then HCM opens the **Physical Mismatch Resolution** dialog to allow the user to get his existing configuration in sync with the MCF (see "Using the master configuration file (MCF)" on page 79).

If the used IODF is MUA-enabled ...

... be aware of how you can work with and how HCM handles such configurations. Refer to "Working with multi-user access enabled IODFs" on page 62.

Opening an existing configuration file with a new IODF

You can deliberately open a configuration file with a new 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 **Open**.

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.

If you exploit the MCF concept (see "Introduction to the master configuration file concept" on page 54 and "Using the master configuration file (MCF)" on page 79), both IODFs involved in the **Open Special...** operation ('old' and 'new' IODF) may have associated MCF data sets. HCM first of all checks whether the old IODF still exists:

- If it does exist, but without associated MCF, it is ignored, just like without MCF exploitation.
- If the old IODF does exist with an associated MCF, then there are the following possibilities:
  - The existing configuration is MCF-enabled: in this case, HCM starts the following processing:

    If the local configuration has set an LC flag, HCM alerts the user and offers the choice to abort the **Open Special...** operation. The user can re-open the configuration with the old IODF and associated IODF in order to propagate the configuration's local changes to the (old) MCF.
  - The existing configuration is not MCF-enabled: in this case, the MCF associated to the old IODF is ignored, and processing is just like without MCF concept.

After HCM has decided about how to process the old IODF, it decides how to continue processing with the new IODF. Table 8 on page 69 lists all possible combinations and describes how HCM reacts while opening the configuration.

Table 8. Opening an existing configuration file with a new IODF

	new IODF with associated MCF	**new IODF without MCF**
existing config is MCF-enabled	If the local configuration has set an LC flag, HCM compares the MCF on the host with the local configuration and offers the **Physical Mismatch Resolution** dialog. Then, if necessary, an IODF resynchronization is invoked.	HCM will MCF-disable the configuration. If required, an IODF resynchronization is started.
existing config is not MCF-enabled	The existing configuration is automatically MCF-enabled. To allow the user to bring physical definitions in sync between MCF and local configuration, HCM starts the **Physical Mismatch Resolution** dialog. Then, if necessary, an IODF resynchronization is invoked.	No special action required. If necessary, an IODF resynchronization is started.

Upgrading configuration files

Using a configuration 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 configuration 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:

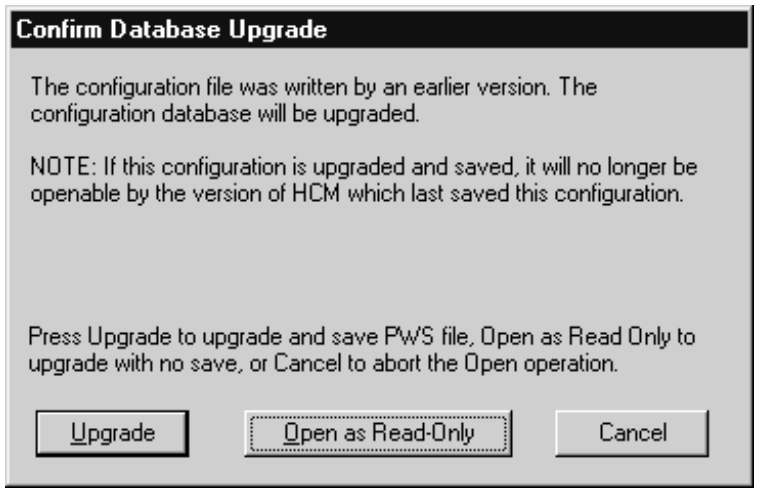


Figure 30. Confirm Database Upgrade

If you select the **Upgrade** button, HCM performs the upgrade operation and then saves the configuration 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 configuration 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 before upgrading, which you can use with the older version or service level of HCM.

Upgrading the associated IODF from version 4 to version 5 format

Starting with z/OS V1R7, HCD uses the new version 5 IODF format. To upgrade both:

- the HCM configuration file to the new data base format,
- the associated version 4 IODF to a new version 5 IODF,

choose one of the following alternatives, depending on your requirements:

Alternative 1: Upgrade the IODF within HCM

1. Open the pre-V1R7 configuration file (**File → Open**). HCM prompts you to either upgrade or open the file as read-only (Figure 30 on page 70). It is recommended to use the **read-only** option, because this choice keeps a backup of the original configuration file. Or you can load a pre-V1R7 IODF into a new or existing, pre-V1R7 configuration file (**File → IODF**). Depending on the size of the IODF, this process can take quite some time because HCD converts the loaded IODF data into the new version 5 format.
2. In both cases, you get an HCD message (CBDG549I), indicating that the associated IODF is temporarily upgraded in-storage only to a version 5 IODF. If you press the **OK** button in this message, HCM loads the configuration file. At this stage, you can perform physical changes only. Logical changes (those that affect the associated IODF) are still prohibited.

3. Use the function **Copy Configuration Files...** from the **File** menu to upgrade both the HCM configuration file and the associated IODF to the new formats. On the **Select Target IODF for Copy** window, select **New...** to create a new IODF. Since the conversion process in step "1" on page 70 releases space within the used blocks, you can usually reduce the allocated space for the new IODF down to the number of used blocks. Save the upgraded HCM configuration file under a new name. You will receive message CBDA483I informing you that the upgraded IODF has been stored in a new IODF.
4. Close the current configuration file and open the upgraded one created in step "3" on page 71. As you used the **Copy Configuration Files...** function, the copied HCM configuration file is already associated with the new version 5 IODF, and hence no resynchronization step is needed in this scenario. You are now ready to update your configuration with logical changes, too.

Note: Next time you build a new production IODF from the upgraded IODF, you may further reduce the number of allocated blocks, since now the IODF data is condensed.

Alternative 2: Upgrade the IODF within HCD

1. You can also use the task **Upgrade I/O definition file to new format** under HCD option **Maintain I/O definition files** to upgrade the version 4 IODF to a version 5 IODF (see *z/OS HCD User's Guide* for a detailed description). This method allows you to condense the IODF in a single step. Depending on the size of the IODF this process - as step "1" on page 70 from alternative 1 above can take quite some time, because HCD converts the loaded IODF data into the new format.
2. In HCM, associate the HCM configuration file or a copy of it with the upgraded IODF (**File → Open Special...**). This way, HCM prompts you for a necessary data base upgrade together with a mandatory resynchronization process which, depending on the IODF size, may take quite some time.

Working with your IODFs and configuration files

You can perform the following tasks with your IODF files using HCM:

- "Copying an existing IODF" on page 71
- "Making a work or a production IODF" on page 72
- "Using the CHPID Mapping Tool Support utility" on page 73
- "Creating work and production IODFs without the associated configuration files" on page 74
- "Exporting and importing IODFs" on page 75
- "Deleting an existing IODF" on page 75
- "Checking a configuration" on page 76
- "Copying a configuration" on page 76
- "Creating work and production IODFs with the associated configuration files" on page 76
- "Compressing configuration files" on page 78
- "Naming convention of your configuration files and IODFs" on page 78
- "Deleting a configuration file" on page 78
- "Saving and closing your configuration" on page 78

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 confirms overwriting the file.

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

If a master configuration file (MCF) is associated to the IODF, it is copied along with the IODF. For information about the MCF concept, refer to "Introduction to the master configuration file concept" on page 54 and "Using the master configuration file (MCF)" on page 79.

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.

- To create a pair of associated work files, the work IODF and the work configuration file, from a production IODF/configuration pair, use action ***File ---> Make Work...***.
- To create a pair of associated production files from a work IODF/configuration pair, use action ***File ---> Build Production...***.

When building 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 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 110). This alternative is further described in "Using the CHPID Mapping Tool Support utility" on page 73.
- **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 about 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 52.
- **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 122).

For any OS configuration that specifies a D/R site OS configuration name as described in "D/R site OS configurations" on page 246, the D/R site OS configuration is automatically generated when building the production IODF.

If the configuration is MCF-enabled...

... the MCF will be copied to the production IODF when it is built. The other way round, the MCF will be copied back to the work IODF when creating a work IODF again from a production IODF.

The master configuration associated with a production IODF is a work configuration file. However, when loading an MCF-enabled production IODF via ***File ---> IODFs... ---> Load***, HCM offers to create a production configuration file (.hcr/.hcrz).

Production configuration files are considered 'read-only snapshots' of a configuration. Therefore, they do not participate in the normal MCF operations: they are not refreshed via download from the host MCF, and they do not update the host MCF via upload.

If a production IODF does not yet have an associated MCF, a user with a production configuration file may enable the configuration for MCF sharing. He may not disable the configuration for MCF sharing, however, for the following reason:

- Other users may own a work configuration file associated with the production IODF. They may apply changes to their configurations (affecting non-IODF objects only). These changes will be propagated to (and from) the MCF as usual.

  The user owning a production configuration file associated with the same IODF is not aware of such changes, because a production configuration file is not refreshed with updates from the MCF. Therefore, HCM does not allow you to set the access mode to ***Single user mode*** for production configuration files, because this would delete the central MCF and other users could not receive these applied changes.

For more information about how to create work and production IODFs, read the following topics:

- "Creating work and production IODFs without the associated configuration files" on page 74
- "Creating work and production IODFs with the associated configuration files" on page 76

Using 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 Resource Link home page (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).

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 110). This utility offers the following functions:

- It exports (downloads) an IOCP file of a selected processor from the host.
- It launches the CHPID Mapping Tool with the exported IOCP file as input. Using the CHPID Mapping Tool, you can assign PCHID values to CHPIDs. The modified IOCP file is saved 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.

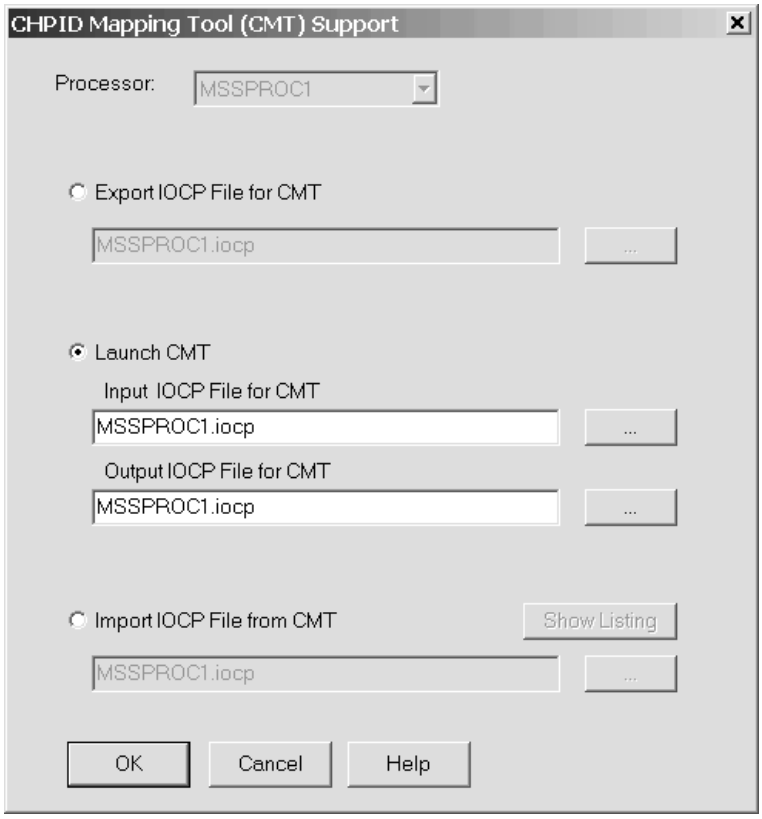


Figure 31. CHPID Mapping Tool Support

Using the dialog shown in Figure 31 on page 73 you can update PCHID values in your configuration performing the following steps:

[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.

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. Following a successful download of the exported IOCP file, select the ***Launch CMT*** radio button. However, this button is only enabled if the CHPID Mapping Tool is correctly installed on your workstation. Note that the feature to be launched from HCM is available for the CHPID Mapping Tool version 4.02 or later.

   In the ***Input IOCP File for CMT*** entry field, specify the IOCP file exported in step "2" on page 74. In the ***Output IOCP File for CMT*** entry field, specify the output file where the CHPID Mapping Tool should write the modified IOCP file with the new PCHID assignment. For IOCP input and output, you can use the same file.

   When you press the ***OK*** button, HCM starts the CHPID Mapping Tool, passing the selected file names as parameters. Now use the CMT to perform the PCHID assignment and save the updated IOCP file. The new IOCP file 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.

   For information about how to use the CMT, refer to its online help and related documentation.
4. Now, back in the ***CHPID Mapping Tool Support*** dialog, 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:

1. If for any reasons for an XMP processor you want to define a channel that is not physically installed on the machine yet, you can use the method of over-defining a channel path as described in "Over-defining a CHPID" on page 125. Thus you can avoid to let the CHPID Mapping Tool run into an error, because it cannot find the PCHID.
2. If the PCHID migration produces an assembler error listing, you can use button ***Show Listing*** to display an assembler listing for error analyzing purposes.
3. When during importing an IOCP data set from the CHPID Mapping Tool, you get a message due to insufficient space allocation, you can use the HCD profile keyword ALLOC_SPACE to overwrite the default space allocation. For more information, refer to the *z/OS HCD User's Guide*.
4. The CHPID Mapping Tool support is not available for configurations with an associated MUA-enabled IODF.

Creating 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 about naming conventions for IODFs on z/VM refer to “Creating a new configuration from scratch” on page 66.

Exporting and importing IODFs

HCM provides functions to export an IODF from the host to the workstation and to import an exported IODF back to the host. You can use these functions for example, to easily transfer IODFs between systems.

Launch the export/import functions by pressing the **Export** or **Import** button from the **IODFs** dialog (see Figure 24 on page 60).

Select an IODF from the list, press **Export**, then use the **Export IODF** dialog, to export an IODF from the host and save it as a file on the PC. In the **Exported IODF file** entry field, HCM proposes a workstation file name for the exported IODF, consisting of the fully qualified IODF name and the extension ***.expiodf***. When you press the **OK** button, HCD saves the exported IODF on the PC and indicates the storage location in a message (this is normally the directory where your configuration files are stored).

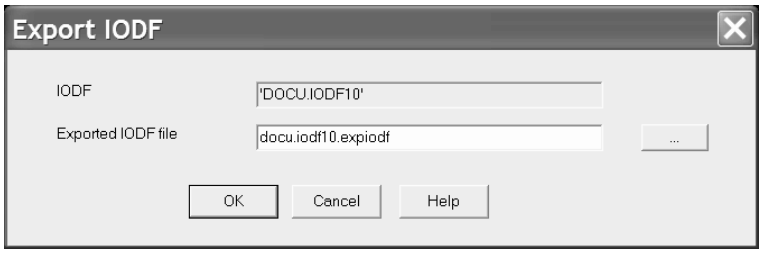


Figure 32. Export IODF

Using the **Import IODF** dialog, you can import an IODF from a file that was previously exported and saved on the workstation. This function builds an IODF on the host, as it is used by HCD. In the **IODF** entry field, you enter a fully qualified IODF name for the imported target IODF on the host. If you selected an IODF from the **IODFs** dialog, this entry field is filled with the name of this selected IODF. If you specify a new name, a new IODF is created on the host.

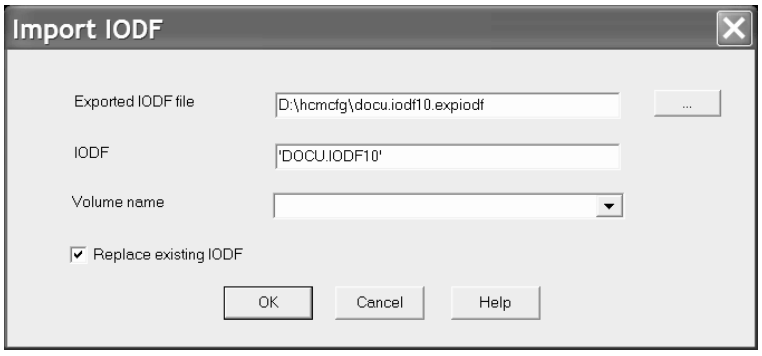


Figure 33. Import IODF

Deleting an existing IODF

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

Checking a configuration

You can check a configuration file for structural consistency. Select ***Check Configuration File...*** from the ***Utilities*** menu. This opens the dialog shown in Figure 34 on page 76.

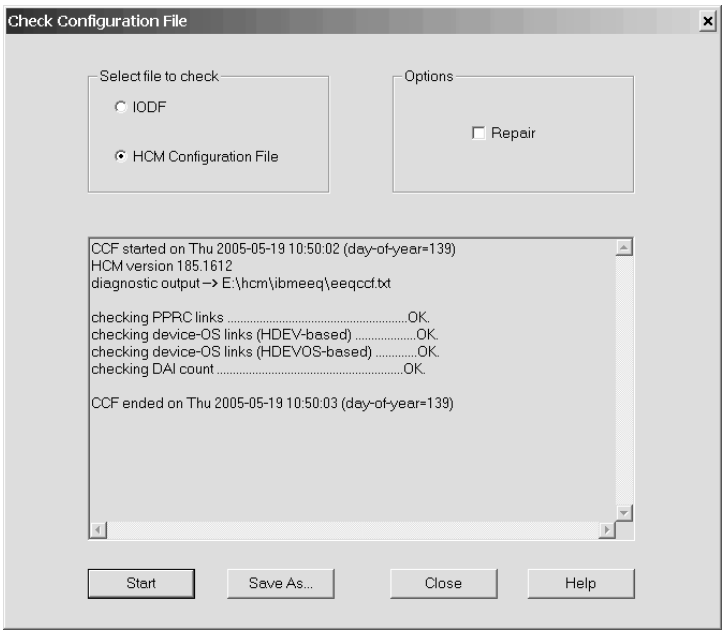


Figure 34. Check Configuration File

Choose which type of configuration file you want to check:

- Select the **IODF** radio button to run a test on the IODF associated with your configuration. This choice is not available in stand-alone mode.
- Select the **HCM Configuration File** radio button to run a test on the configuration file on the workstation. This option is also available in stand-alone mode, however, the utility can check and repair only inconsistencies in the configuration file that are not related to the associated IODF.

Select the **Repair** option to attempt a repair of any structural inconsistencies found in either the configuration file or the associated IODF. You cannot use this option when checking a production IODF.

Press the **Start** button to run the test. Messages resulting from the check on the host or on the workstation are displayed in the dialog's output box. You can use the **Save As** button to store the messages in a file. If you run the test on a configuration file on the workstation, HCM may write further diagnostic output into a file named **eeqccf.txt** which you can find in the HCM installation directory. If you need to contact IBM support for help, you can send the complete output produced by the checking utility.

Copying a configuration

You can backup the configuration file and its IODF before making changes to them. Use the **Copy Configuration Files...** pull-down choice of 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 makes copies of the current files, leaving the original files open for you to continue working. If you have changed the IODF or configuration file, use the **Save** command in the **File** menu to save the changes before using the **Copy Configuration Files...** function.

Creating 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/.hcrz file is saved under the name supplied in the **Save Production File** dialog.

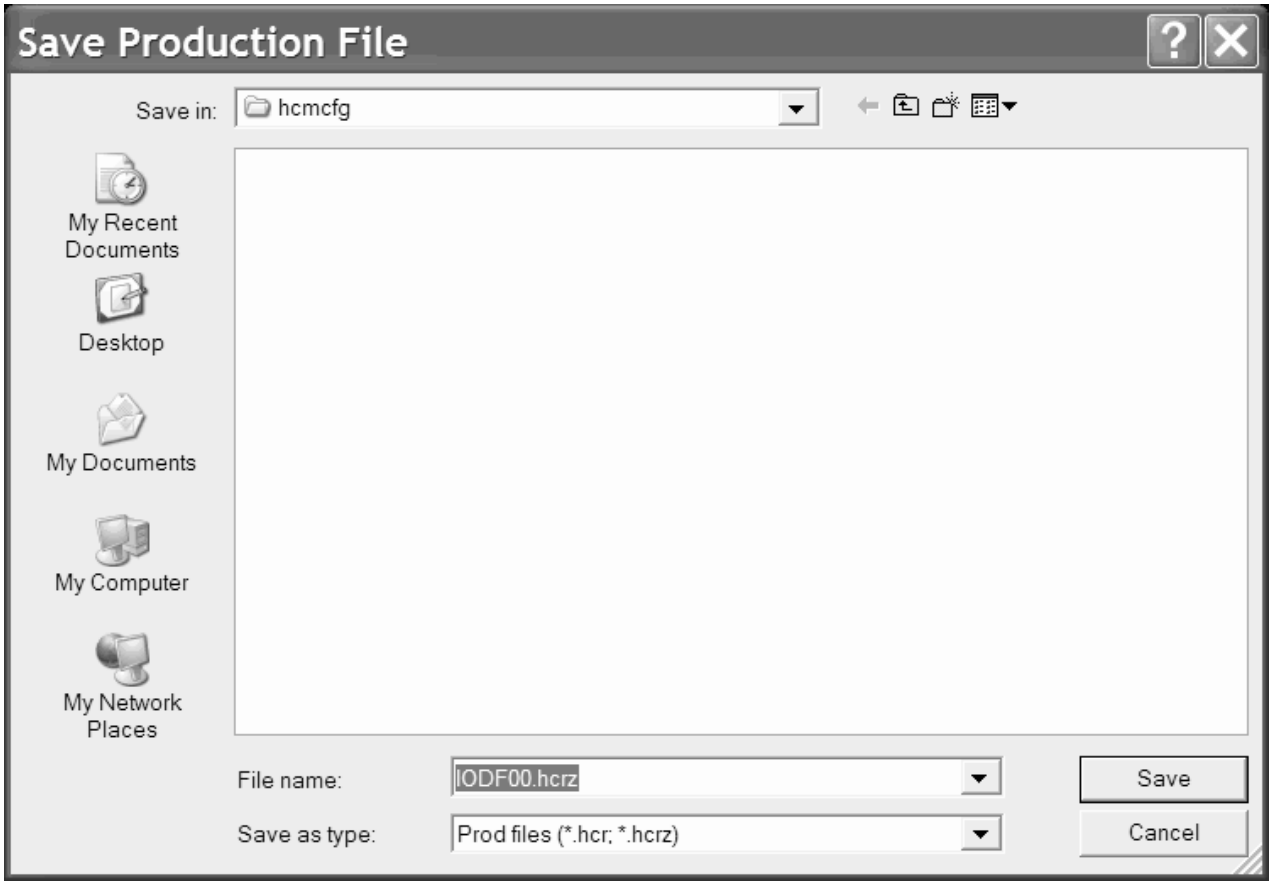


Figure 35. Save Production File dialog

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 is 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 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.

Compressing configuration files

Using ***Options ---> File Compression...*** from the menu bar, you can choose to store your HCM configuration in a compressed format. This choice is recommended, because compressed files consume less disk space and provide a better performance while opening and saving. Compressed work configuration files have a file extension of ***.hcmz*** while uncompressed ones have the traditional extension ***.hcm***. Likewise, compressed production configuration files have a file extension of ***.hcrz*** while uncompressed ones have the extension ***.hcr***.

If you compress an MCF-enabled configuration, the associated MCF is also compressed.

HCM can read and load both the uncompressed and the compressed formats. The file compression option is checked only when storing the files on the disk. The default format is uncompressed.

Note: Compressed HCM configurations cannot be opened with HCM releases prior to z/OS V1R9. To uncompress a configuration, first select the option to not compress configurations, then open the compressed configuration, apply at least one minor, unimportant change, and save the configuration.

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 PC 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, an iodf92.hcr file (or iodf92.hcrz for compressed files) on the PC. When IODF92 requires modification, a work IODF and a work PC file are generated. The IODF receives the name IODF93.WORK and the PC file receives the name iodf93.hcm (or iodf93.hcmz). Once the changes are made, new production versions are built with the names IODF93 and iodf93.hcr (iodf93.hcrz).

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 configuration 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 configuration file and IODF with matching tokens.

After having updated the configuration, the **Close** command saves both the IODF and the configuration file, then closes both files. When you close the IODF with **Close** or **Exit**, the **Activity Log** dialog will appear if activity logging has been enabled for the IODF (see “Creating a new configuration from scratch” on page 66).

Closing an MCF-enabled configuration also updates the associated MCF on the host.

You can enable ***automatic activity logging*** by entering the following keyword into the HCD profile:

```
CHANGE_LOG = YES
```

With this setting, HCD uses a subset of the change log entries to generate automatic entries into the activity log for updates on configuration data, like for example for add/delete or connect/disconnect activities. These entries are HCD proposals which you can edit.

Note: If on HCD, for any reason, the change log data set (CHLOG) is not accessible, no automatic activity log entries can be created.

For more information, refer to the *z/OS HCD User's Guide*.

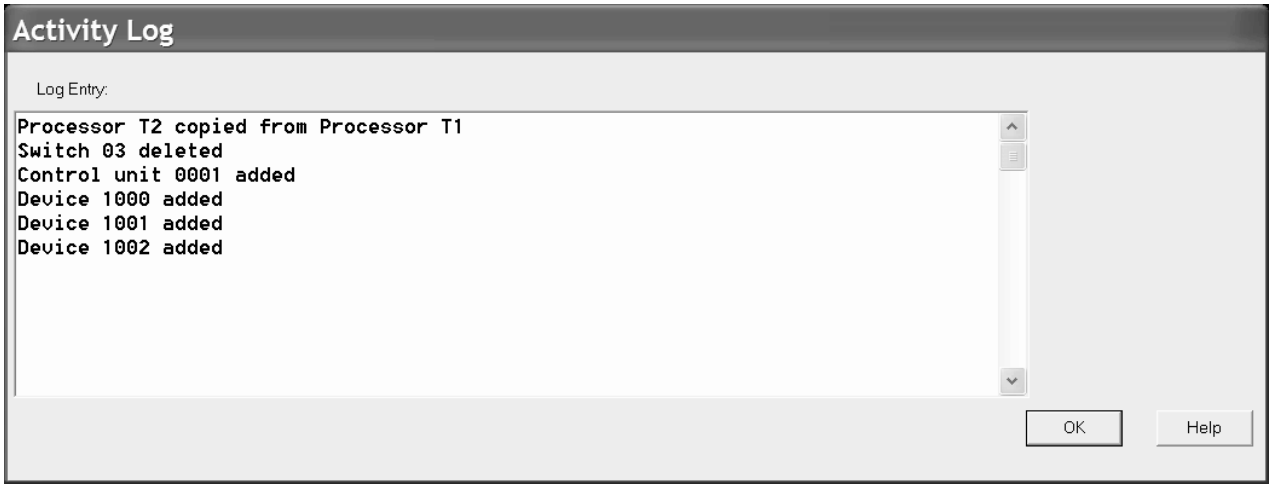


Figure 36. Activity Log dialog

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 66 for this file's naming convention.

File → Exit performs a **Save** before exiting HCM.

If HCM terminates because of a host abend, a loss of network connection, or an HCM abend, then the next time you open your configuration 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.

Using the master configuration file (MCF)

The master configuration file (MCF) is an HCM configuration stored on the host as a central shared repository. The MCF concept is introduced in "Introduction to the master configuration file concept" on page 54 and brings along the following main advantages:

- Several HCM users can work cooperatively and safely on a single configuration, sequentially sharing the configuration in connected mode.
- There is no longer the need to keep this critical data on workstations, maybe even distributed as multiple local copies across your installation, but you can maintain a single consistent data set on the mainframe.

Sharing an HCM configuration via MCF is recommended when two or more users need to make changes to the configuration. Although in cases of conflict, MCF sharing provides a **Physical Mismatch Resolution** dialog (see "How to resolve physical mismatches in a shared configuration" on page 86), the respective tasks of the users should be carefully planned and agreed upon. This eases later mismatch resolutions.

The local configuration file and the master configuration file both carry a timestamp to allow HCM to detect changes done to non-IODF objects. Resynchronization and physical mismatch resolution is then

based both on the LC flag and the various timestamps: the MCF physical timestamps, timestamps of the local configuration and on the IODF timestamp.

Using the HCD profile keyword MCF_VOL lets you specify a volume serial number where to store the MCF on the host, refer to the *z/OS HCD User's Guide*.

Here is an outline of the MCF mode of operation:

- If you work in connected mode, the MCF is updated and stored on the host whenever you close your configuration.
- If you work in stand-alone mode, you work on a local copy of your configuration. HCM cannot synchronize with the host-based MCF immediately, but sets a so-called ***LC flag***, indicating that the configuration was locally changed.
- During ***Open configuration*** processing, HCM compares the timestamps of the local configuration and the master configuration and distinguishes the following scenarios:
  - If the timestamps of the local and the master configuration are the same, your configuration is opened as usual.
  - If the local configuration is older than the MCF, and no local changes (no LC flag) are found in the configuration, then, after the user's confirmation, HCM overwrites the local configuration by downloading the MCF from the host. This ensures, that logical changes to the IODF and physical changes to the MCF are propagated to the current user.
  - If HCM encounters the LC flag to be set, then the following cases are distinguished:
    1. Timestamps of MCF and local configuration are the same:

       This is the case, if a user had accessed the MCF in connected mode and ended the session with MCF, local configuration and IODF in sync. Later, the same user has updated the local configuration in stand-alone mode. Even later again, the same user accesses the configuration in connected mode, without any other user having accessed the configuration in the meantime. Opening processing continues as usual and only at closing time of the configuration, HCM updates the MCF on the host with all of the user's changes (from the previous stand-alone and the current connected session).
    2. Timestamps of MCF and local configuration are different:

       This is the case, if another user has accessed the MCF in connected mode. In this case, HCM opens the **Physical Mismatch Resolution** to let the current user resolve physical mismatches due to the changes in his local configuration.

Note: The MCF feature is new with z/OS 1.9. Its exploitation is optional. Without using this feature, processing of HCM configurations continues to work as in previous releases.

Enabling/disabling a configuration for MCF-sharing

With action **File ---> Set Access Mode**, you invoke a corresponding dialog (see Figure 37 on page 81) where you can enable the current configuration to use the MCF concept by selecting the ***Master configuration file mode*** radio button.

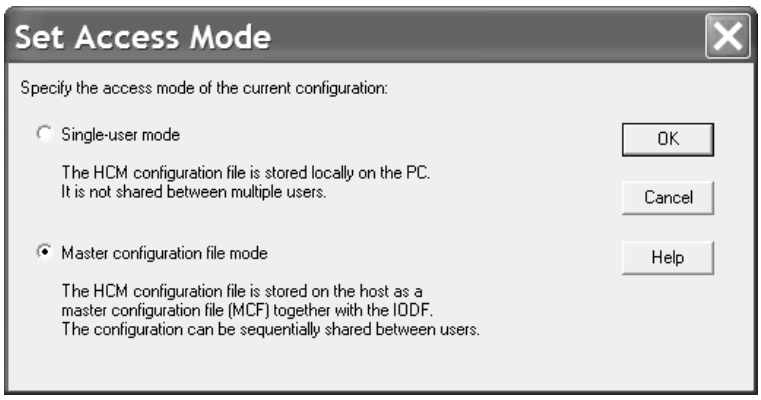


Figure 37. Set Access Mode

When you mark a configuration as MCF-enabled, HCM creates the MCF and uploads it to the host when you close the configuration:

- In z/OS, the MCF is stored on the host in a data set `'hlq.IODFnn….MCF'` (that is, the name of the associated IODF + suffix `.MCF`).
- In z/VM the MCF is stored in a CMS file where the name consists of the CMS file name of the IODF and one of the following file types:

  WORKMCF for the master configuration file of a work IODF
  PRODMCF for the master configuration file of a production IODF
  BACKMCF for the master configuration file of a backup IODF

By selecting the ***Single-user mode*** radio button, you can MCF-disable the current configuration. This operation deletes the existing MCF from the host.

Granting access to the MCF on the host

In your installation, you may have several users with different roles concerning the task to manage I/O configurations. These users must have different access rights to the IODFs and the MCF data sets on the host. This section offers a proposal for a security set up based on RACF or a different security product for the following user groups:

- **System programmer**: responsible for all of an installation's I/O configurations and must have comprehensive access rights on IODFs and MCFs.
- **I/O Configuration manager (class A)**: responsible for physical configurations, like cabling, and the consistency of MCF-enabled configurations, but not authorized to perform logical changes like creating or configuring processors. Is authorized to enable/disable configurations for being shared and authorized to update MCFs. Also must be able to retrieve any information (physical and logical) about configurations.
- **I/O Configuration manager (class B)**: must be able to perform physical changes on MCF-enabled configurations and therefore needs authorization to update (upload) MCFs on the host. Must be able to retrieve any information about configurations, but is not authorized to enable/disable MCFs.
- **System operator**: must be able to retrieve any information about configurations, but is not authorized to perform any updates.

Table 9 on page 82 shows the I/O configuration tasks that users of the described groups must perform. Table 10 on page 82 shows the proposals on how to specify the appropriate staged access rights.

Table 9. I/O configuration: responsibilities and tasks				
Configuration task	**System programmer**	**I/O Configuration manager (class A)**	**I/O Configuration manager (class B)**	**System operator**
Perform all required I/O configuration tasks: update all IODFs and MCFs	yes	no	no	no
Apply physical changes only (includes physical mismatch resolution): update MCFs	yes	yes	yes	no
Observe consistency among shared physical configurations: enable/create and disable/ delete shared MCFs on the host	yes	yes	no	no
Viewing I/O configurations: load IODF/ download MCF to PC	yes	yes	yes	yes

Table 10. I/O configuration: required access rights and tasks				
Data set specification	**System programmer**	**I/O Configuration manager (class A)**	**I/O Configuration manager (class B)**	**System operator**
HLQ.IODF%%.**	alter	read	read	read
HLQ.IODF%%.**.MCF HLQ.IODF%%.**.MCF.CLUSTER	alter	alter	update	read

Note: Applying many physical updates to a configuration may require to enlarge an MCF data set by deleting the old and allocate a new data set on the host. To allow users to insert a great amount of physical configuration information, you may in advance enlarge the MCF data set beyond the space that would be allocated per default. Use the HCD profile option `MCF_EXTENSION` to specify the percentage of additional space that is to be allocated when defining an MCF data set. For a detailed description of the `MCF_EXTENSION` keyword, refer to the *z/OS HCD User's Guide*.

How to work with MCF-enabled configurations: Scenarios

This section presents the following usage scenarios for working with MCF-enabled configurations:

- "Loading an IODF with associated MCF" on page 82
- "Opening an MCF-enabled configuration" on page 83
- "Opening a configuration that was MCF-disabled by another user" on page 83
- "Processing an MCF-enabled configuration stand-alone and connected" on page 83
- "Version checking: using an old HCM version" on page 85
- "Version checking: using an old HCD version" on page 85

Loading an IODF with associated MCF

Pressing the **Load...** button on the **IODFs...** dialog (Figure 24 on page 60), loads an HCD IODF onto your workstation. If you want to load an existing IODF as a starting point for a new configuration, an MCF data set may already be associated with the specified IODF. In such a case, you inherit the HCM configuration file from the MCF according to the following scenario:

Scenario: Loading an IODF with associated MCF

```
- User A creates a new IODF for a new configuration or works with an
  existing IODF/configuration, makes a few logical changes that affect the IODF,
  and also makes some physical-only changes (for example, create a
  cabinet/general box, named cables, and so on).
- User A MCF-enables the new configuration and closes it.
  The MCF is uploaded to the host.
- User B selects File ---> IODFs ---> Load…
  for the IODF just saved by user A and associates it to a new configuration.
- Result: The HCM configuration file for user B is created from the MCF, and user B
  sees the physical-only changes done by user A.
```

Opening an MCF-enabled configuration

The HCM **Open...** operation on existing configuration files checks whether this configuration is MCF-enabled. If an associated MCF is found, HCM checks the consistency of the user's> local configuration with the MCF by comparing the timestamps. If they are in sync, the opening processing continues as usual, with an IODF resynchronization, if required.

When HCM finds a more recent MCF than the user's> local configuration (and there are no stand-alone mode changes in the local configuration), the MCF automatically replaces the user' configuration file after his confirmation.

Scenario: Opening an MCF-enabled configuration

```
- Steps from scenario Loading an IODF with associated MCF are completed:
  ---> User B is accessing the configuration/IODF
- User B makes further logical and physical-only changes
- User B closes the configuration, MCF is uploaded to the host
- User A opens his local configuration (in connected mode):
  result: local configuration is outdated, therefore:
  1) The MCF is downloaded and replaces the local configuration.
  2) The replaced local configuration is opened.
    Since user B left the MCF in sync with the IODF, no IODF
    resynchronization is required for user A.
  User A now sees additional physical-only and logical changes done by user B
```

Opening a configuration that was MCF-disabled by another user

When one HCM user MCF-disables a configuration, another HCM user of the same configuration will be notified when he connects to the associated IODF the next time. This user's local configuration is also set into MCF-disabled mode.

Scenario: Opening a configuration that was MCF-disabled by another user

```
- Steps from both scenarios are completed:
         Loading an IODF with associated MCF
         Opening an MCF-enabled configuration
  ---> User A is accessing the configuration/IODF
- User A makes additional changes: both logical and physical-only
- User A MCF-disables the configuration (MCF data set is deleted on the host)
  and closes the configuration
- User B opens his local configuration (in connected mode)
  result: HCM issues a message to user B:
  "No Master Configuration File was found on the host for your configuration.
  It is therefore currently marked as disabled for MCF sharing.
  You may want to re-enable MCF sharing after your configuration is loaded."
- Standard resync processing without MCF-enabling takes place;
  User B sees IODF changes done by user A, but no physical-only changes.
  User B may MCF-enable his configuration again.
```

Processing an MCF-enabled configuration stand-alone and connected

Besides the sequential sharing in connected-mode, the MCF concept also allows several HCM users to work stand-alone on different local copies of a single configuration. When the users later connect to the IODF, HCM detects inconsistent or conflicting changes and starts the **Physical Mismatch Resolution** dialog. This dialog will list these conflicts and allow the user to decide which variant to keep. When

mismatch resolution and resynchronization are complete, the resulting consistent configuration is sent to the host as the new MCF, ready and available for the next user.

When an HCM user **A** edits an MCF-enabled configuration in stand-alone mode, HCM sets the ***LC flag*** to indicate that the configuration was locally changed. User **A** can make physical changes to the configuration in one or more stand-alone sessions.

When user **A** opens the configuration in connected mode the next time, HCM checks the LC flag. Provided, no other user changed the MCF on the host in the meantime, the MCF on the host will be automatically refreshed with user **A**'s local configuration (when the configuration is closed).

Another possibility is that another user **B** has applied changes in connected mode in the meantime. That is, the MCF data set on the host has been updated with these latest changes. When user **A** re-opens the configuration, HCM checks the LC flag and starts the **Physical Mismatch Resolution** dialog.

Here are the scenarios in detail:

Scenario: Reconnect after stand-alone changes with logical changes made in the meantime by another HCM user (without mismatch resolution or IODF resynchronization)

```
- User A opens a local copy of an MCF-enabled configuration in stand-alone mode,
  which is in sync with the host MCF and IODF.
- User A makes physical-only changes (for example, create general box, named cable),
  closes the configuration and exits HCM.
  HCM sets the LC flag indicating local modifications.
- User A opens the configuration again in connected mode.
- Though the LC flag is set, HCM recognizes that no other user has changed the
  MCF in the meantime. HCM can postpone the update of the MCF until user A
  ends the current connected session.
- No IODF resynchronization is required and the session proceeds normally.
- User A makes additional changes, then closes the HCM configuration.
- HCM updates the MCF with previous and additional changes done by user A
  and clears the LC flag.
- Later, user A opens the configuration again in connected mode:
  Supposed, another user B has modified the IODF meanwhile using HCM and
  therefore left an updated MCF on the host. As user A's configuration
  has no longer set the LC flag, it is just refreshed by an MCF-download from
  the host. Therefore,  user A is completely in sync with the IODF and the MCF,
  and neither an IODF resynchronization nor a physical mismatch resolution
  is performed.
```

Scenario: Reconnect after stand-alone changes with logical changes made in the meantime by another HCD user (without mismatch resolution, with IODF resynchronization)

```
- User A opens a local copy of an MCF-enabled configuration in stand-alone mode,
  which is in sync with the host MCF and IODF.
- User A makes physical-only changes, closes the configuration and exits HCM.
  HCM sets the LC flag in user A's configuration.
- HCD user B opens the IODF using HCD and modifies IODF objects (logical changes).
  Note: Updates on the IODF using HCD on the host do not touch the associated MCF.
- User A opens the configuration again in connected mode.
- Though the LC flag is set, HCM recognizes that no other user has changed the
  MCF in the meantime. HCM can postpone the update of the MCF until user A
  ends the current connected session.
- However, HCM recognizes that the IODF has been changed by HCD user B,
  and therefore, IODF resynchronization is started.
- User A makes additional changes, then closes the HCM configuration.
- HCM updates the MCF with previous and additional changes of user A
  and with logical changes applied by HCD user B and clears the LC flag
  in user A's configuration.
- IODF and MCF are now consistent and no other updates are pending.
```

Scenario: Multiple logical changes by HCM/HCD users on an MCF-enabled configuration (with mismatch resolution and IODF resynchronization)

```
- User A opens a local copy of an MCF-enabled configuration in stand-alone mode
  which is in sync with the host MCF and IODF.
- User A makes physical-only changes, closes the configuration and exits HCM.
  HCM sets the LC flag in user A's configuration.
- User B connects to this IODF with his local configuration and makes several
  physical and logical changes. User B closes the configuration.
- HCD user C opens the IODF, modifies IODF objects (logical changes)
  and closes the IODF.
```

```
  Note that the associated MCF on the host is not updated by HCD.
  Only HCM can update an MCF in connected mode.
- User A opens the configuration again in connected mode.
  As another user B meanwhile has applied physical and logical changes
  to the configuration, HCM shows the explanatory message:
   "Your configuration is marked as modified in stand-alone mode.
   The Master Configuration File contains changes done in connected mode.
   You can resolve mismatches in the following dialog."
- HCM downloads and compares the MCF with the local configuration file of user A.
- HCM invokes Physical Mismatch Resolution dialog; user A resolves conflicts.
- As the MCF configuration (left on the host by user B)
  is still not in sync with the latest version of the IODF as
  modified by user C, HCM invokes IODF resynchronization, so that
  the following status is achieved:
  - non-IODF objects are as defined by physical mismatch resolution by user A
  - IODF objects are as defined by HCD users B and C
  - When user A closes the configuration,
HCM  will upload the consolidated MCF
```

Scenario: Shared logical changes on an initially MCF-disabled configuration

```
- HCM users A and B are both keeping local copies of a configuration associated
  with the same IODF. Both local configurations are not yet MCF-enabled.
- User A connects to this IODF, then MCF-enables his configuration and exits.
  An MCF which is in sync with the IODF is created and uploaded to the host.
  From now on, all users of this IODF will be informed about any changes applied
  to the MCF.
- User B connects to this IODF with his configuration. HCM detects that
  the configuration has been MCF-enabled and issues the following message:
    "Your configuration was previously disabled for MCF sharing.
    A Master Configuration File now exists for the associated IODF.
    Your configuration will be enabled for MCF sharing now.
    You can resolve mismatches in the following dialog."
  HCM checks if the Physical Mismatch Resolution dialog needs to be
  invoked to let the user resolve physical mismatches.
- User B can now work on the configuration as usual.
  The resulting updated configuration will be uploaded as the new MCF when user B
  closes the configuration.
```

Version checking: using an old HCM version

HCM users can connect to an MCF-enabled configuration using an older HCM version without MCF support. However, HCD alerts these users that the configuration is MCF-enabled and that the HCM version being used is outdated.

Scenario: Connecting to an MCF-enabled configuration using an HCM version earlier than z/OS 1.9

```
- A user accesses an MCF-enabled configuration in connected mode, using
  an HCM version earlier than z/OS 1.9
- Result: Open action succeeds, but HCM gives a warning message:
  "The configuration you are opening has an associated master configuration
  file (MCF). You are using an old HCM version without MCF capability.
  Physical modifications (i.e., non-IODF changes like cabling, cabinets etc.)
  previously done by others will not be visible to you.
  Physical modifications done by you in this session will not be propagated
  to the MCF. Please continue only if this is acceptable."
```

Version checking: using an old HCD version

HCM users can connect to an MCF-enabled configuration on a host running an HCD version without MCF support. An MCF-capable HCM detects this version mismatch and alerts the user.

Scenario: Connecting to an MCF-enabled configuration using an HCD version earlier than z/OS 1.9

```
- A user accesses an MCF-enabled configuration in connected mode, using an HCM
  version with MCF capability to connect to an IODF on an HCD version earlier
  than z/OS 1.9
- Result: HCM displays a message:
  "The configuration you are opening indicates that an associated master
  configuration file (MCF) on the host is expected. HCM cannot use
  (or even verify the existance of) the MCF because the HCD (host) version used
  does not yet support MCFs. Physical modifications (i.e., non-IODF changes like
  cabling, cabinets etc.) done by others will not be visible to you. Physical
  modifications done by you in this session will not be propagated to the MCF.
```

```
  Do you wish to continue?"
- A) User answers "no":
  opening the configuration is aborted (the IODF will not be opened).
- B) User answers "yes":
  processing the configuration continues without exploiting the MCF capability.
  However, HCM sets the LC flag for local changes, just like in stand-alone mode.
  This will trigger the Physical Mismatch Resolution dialog later, if the user
  connects to an MCF-enabled HCD version (z/OS 1.9 or later).
```

How to resolve physical mismatches in a shared configuration

As indicated in several scenarios from "How to work with MCF-enabled configurations: Scenarios" on page 82, physical mismatches in configurations can occur if users access different local copies associated with the same IODF. If HCM in connected mode detects mismatches between a user's local configuration and the MCF data set on the host, the **Physical Mismatch Resolution** dialog (Figure 38 on page 86) is started to allow the user to get his local configuration in sync with the MCF.

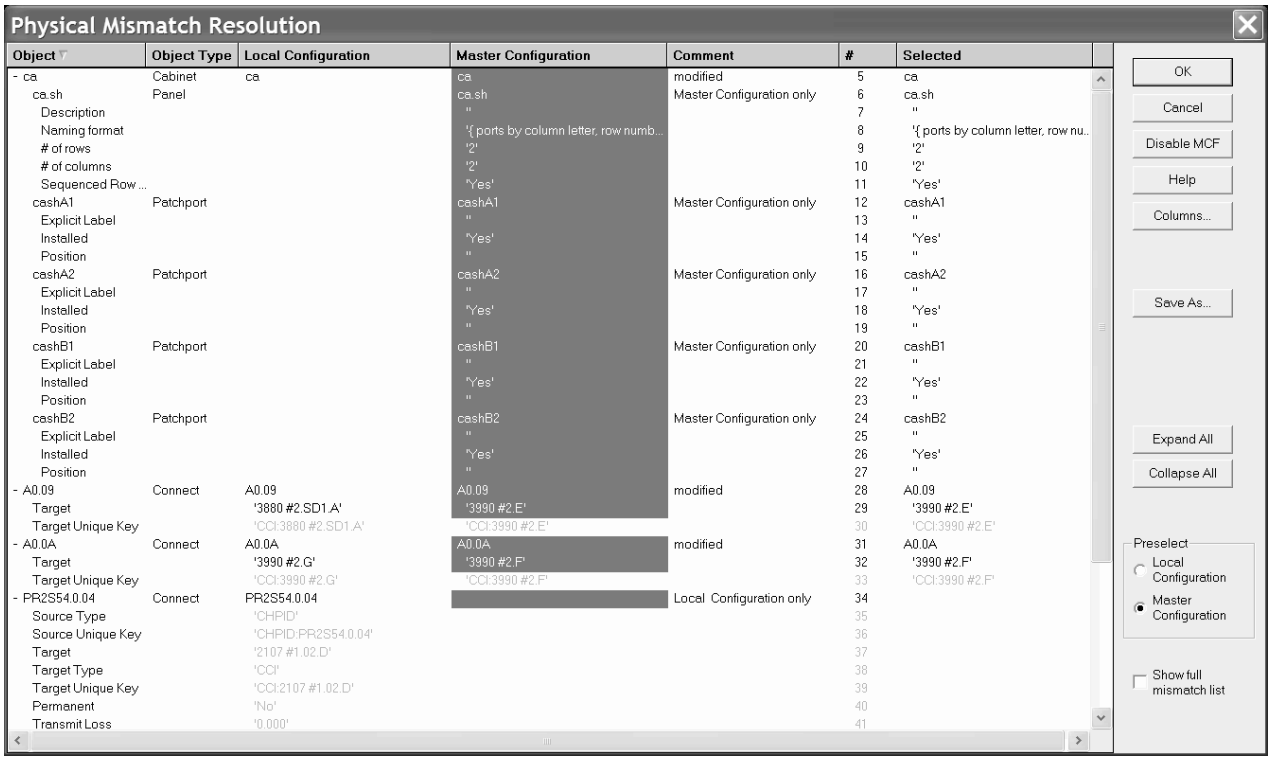


Figure 38. Physical Mismatch Resolution

The **Physical Mismatch Resolution** dialog displays the mismatches between the local configuration and the master configuration downloaded from the host. Each row of the dialog displays one mismatch. By clicking either on the local variant in column ***Local Configuration*** or on the MCF variant in column ***Master Configuration***, users can now select for each mismatch, which variant should be kept in the final configuration.

By default, the master configuration variant of a mismatch is preselected.

Double-click on the plus (+) sign of an object or connection to see the individual attributes of that object/connection.

Only the complete object can be selected in the resolution dialog, that is, you cannot select individual attributes of objects. Also, you can only select complete connections.

In our example from Figure 38 on page 86, if the user presses the ***OK*** button, the master configuration on the host will be updated with the local changes that had been performed in stand-alone mode. HCM displays a textual summary of the physical mismatch resolution as selected by the user. The summary is divided into the following sections:

- ***Resolution of the following mismatches was successful:*** lists mismatches that were successfully resolved.
- ***Resolution of the following mismatches failed:*** may list mismatches that cannot be resolved, for example, because the user made conflicting selections.
- ***The following mismatches are not resolvable by MCF physical mismatch resolution:*** may list mismatches for which no resolution was attempted, for example, because they affect attributes of the configuration that are of a logical nature and therefore are stored in the IODF.

If you mark the ***Show full mismatch list*** check box, you can also see potential logical mismatches. However, you cannot resolve these mismatches in this dialog. With this check box unmarked, only physical mismatches that you can resolve with this dialog are displayed.

Note:

Comparing MCFs for mismatch resolution is based on the contents of files created by the HCM **Export Data...** function (see Chapter 14, “Importing and exporting data,” on page 333). Therefore, messages EEQ0260E and EEQ0261E that may appear while exporting the involved data, can also appear while processing MCFs. Just ignore the messages.

Recommendations and guidelines for mismatch resolution

The mismatch resolution dialog allows you to resolve mismatches that occur if several users modify multiple local copies of the same configuration either in connected mode or in stand-alone mode. Depending on the changes, the **Physical Mismatch Resolution** dialog may get very voluminous and requires a careful selection of the changes that should be taken into the MCF. Therefore, it is ***strongly recommended*** to plan the changes between the co-workers. The following guidelines and recommendations assist you in taking advantage from working in parallel using the MCF mismatch resolution.

To avoid rework after the mismatch resolution, note the following guidelines:

- Rename complete objects, for example, controller labels or string labels, only in connected mode and when no other configuration work is done in parallel. The same is true for the renaming of panel IDs or device types. This is recommended because the comparison of configurations is based on the contents of the key columns in the exported files from the HCM **Export Data...** function. Therefore, a renamed object is interpreted as both a deleted old and an added new object, thus resulting in an unexpected amount of mismatches.
- Converter labels are implicitly assigned with a name that includes a sequence number when connecting a parallel to a serial interface. Since the **Physical Mismatch Resolution** dialog shows a modification for matching labels, it is recommended to apply unique names to the converter labels in a configuration.
- Avoid to perform changes on controllers or strings when working in parallel or in stand-alone mode, for example, rearranging controller channel or device interfaces, or reassigning units or devices. HCM may not be able to resolve such a mismatch properly. The same is true for modifying connections between controllers and strings, as well as for changing the physical description file (PDF) of a controller.
- Before working in stand-alone mode, first connect to the host to download the latest version of the MCF configuration.
- After having finished work in stand-alone mode, do not defer reconnecting to the host in order to put the changes into the MCF.

Therefore, when performing mismatch resolution, respect the following recommendations:

- If a connection is selected, make sure that also the connecting objects (with matching interface types) are included in the target configuration. Otherwise, the connection cannot be established.
- Carefully review the **Summary** page displayed by the **Physical Mismatch Resolution** dialog, especially for mismatches that are not successfully resolved. If this is due to an inconsistent selection, press the ***Discard Resolution*** button and correct the current selection.

Consider, for example, a straight connection between a controller and a channel path: the controller and the connection may exist in the master configuration, but not in the local configuration. This results in one mismatch for the controller and a separate mismatch for the connection. The user might now select the connection to be included into the local configuration, but not the controller. This selection is inconsistent: when trying to establish the connection, HCM fails on this particular mismatch because one of the two objects to be connected is not present in the local configuration (the controller that was de-selected). This is listed on the dialog's **Summary** page as a failed mismatch with the reason indicated such as `'object not found <controller interface name>'`.

Note: Changes to the position attribute of patchports and converters are not resolved during mismatch resolution. Subsequent manual adjustment may be required.

The following changes may be done in stand-alone mode and later included smoothly in the MCF:

- creation, deletion and modifications of physical-only connections between processors, switches, crossbar switches, and controllers
- inclusion and modification of cables in the configuration
- definitions of cabinets and general boxes
- inclusion of patchports into physical connections
- definition and modifications of user fields

You can use mismatch resolution for efficient work in the following scenario of a configuration file that includes two sites, **A** and **B**:

1. The system programmer responsible for the I/O configuration of site **A** performs changes working directly with the MCF in host-connected mode.

   At the same time, the hardware planner of site **A** defines cables for the physical connections in stand-alone mode. Afterwards, the hardware planner connects to the host and updates the MCF via the **Physical Mismatch Resolution** dialog.
2. Now, the system programmer who is responsible for the I/O configuration of site **B** performs changes in connected mode, based on the previous changes that are contained in the MCF. The hardware planner of site **B** may perform physical-only changes in stand-alone mode and include them later in the MCF via mismatch resolution when the system programmer has finished his work.

How to work with configuration packages

Large enterprises may want to manage their I/O configurations from centrally administered large master IODFs. Nevertheless, there might be the need to transmit a subset of a master IODF for use on a certain target system. To preserve this benefit of keeping a single repository for all I/O definitions, users can build subset IODFs from a centrally administered master IODF. Such a subset IODF is called a configuration package. Configuration packages allow selective distribution of I/O data to target systems, while keeping the processor token in sync.

Up to z/OS 1.9, the concept of configuration packages had only been implemented in HCD. Starting with z/OS 1.10, HCM also provides an interface to the functionality of configuration packages, except for the merge and replace functions, which remain available in HCD only. Starting with z/VM 6.2, z/VM users can also exploit this functionality using the HCM dialogs described in this topic.

Working with configuration packages is similar in HCD and HCM. In case you need further information, it is recommended to refer to the following topics in the *z/OS HCD User's Guide*:

- The master IODF concept
- Work with configuration packages
- Problems with the Transmit Configuration Package action
- User authentication for unattended mode

To work with configuration packages in HCM, from an open configuration, select ***Configuration Packages...*** from the ***Edit*** menu to display the ***Configuration Packages*** dialog. This dialog displays a list of defined packages. From here, you can perform all applicable actions on selected configuration packages.

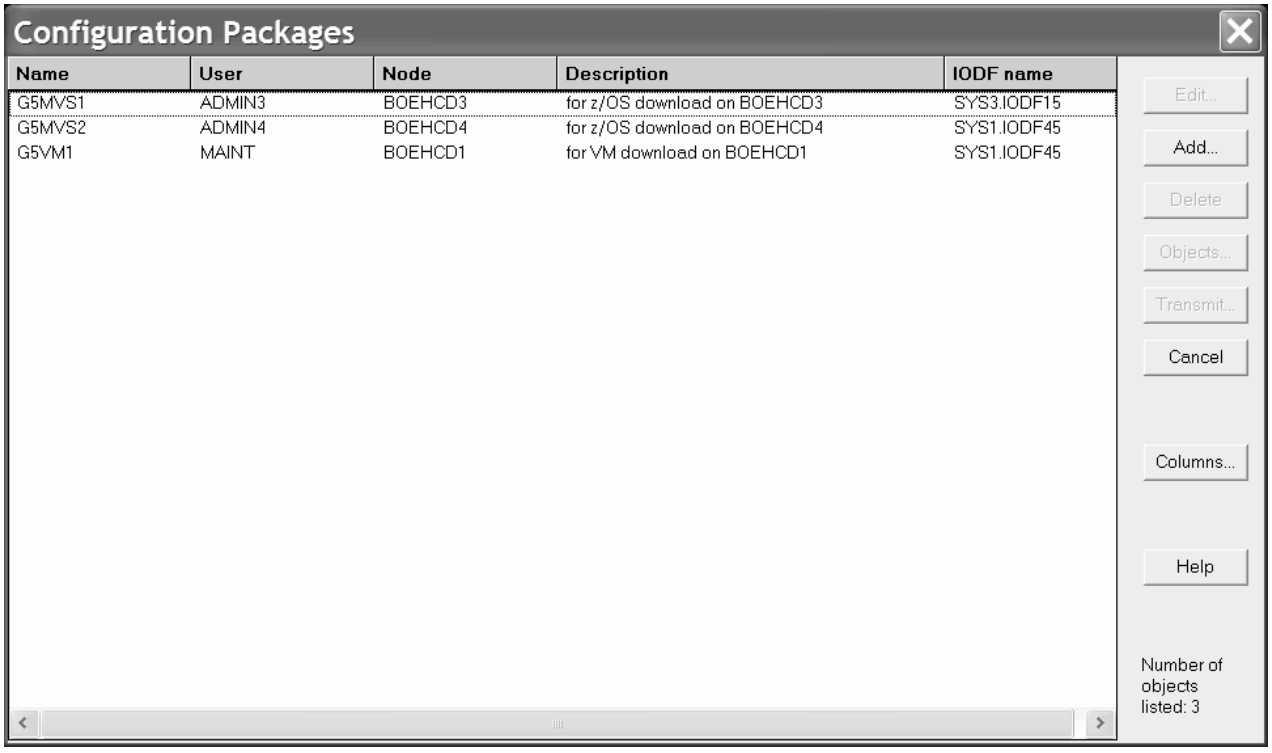


Figure 39. Configuration Packages

Some brief rules for working with configuration packages:

- You can create or edit configuration packages only from an accessed work IODF.
- You can transmit configuration packages only from an accessed production IODF.
- You can transmit configuration packages only if they contain at least one processor or operating system configuration.
- Before transmitting an edited configuration package, you must rebuild the production IODF from the master IODF.

How to create and edit a configuration package

To create a configuration package as part of a (master) IODF, open a configuration with this associated work IODF, open the ***Configuration Packages*** dialog and press the ***Add...*** button. HCM displays the ***Create Configuration Package*** dialog. For just defining a package, only a name for the package and the target IODF data set name are required. For later transmitting the package, the target user ID and node, and the target IODF data set name are required.

Note that for z/OS, HCM inserts the HLQ of the accessed work IODF in front of the specified target IODF if you do not enclose the target IODF name in quotation marks.

To edit a defined configuration package, select this package and press the ***Edit...*** button in the ***Configuration Packages*** dialog to display the ***Edit Configuration Package*** dialog, which is similar to the ***Create Configuration Package*** dialog. You can edit all package attributes except for the package name.

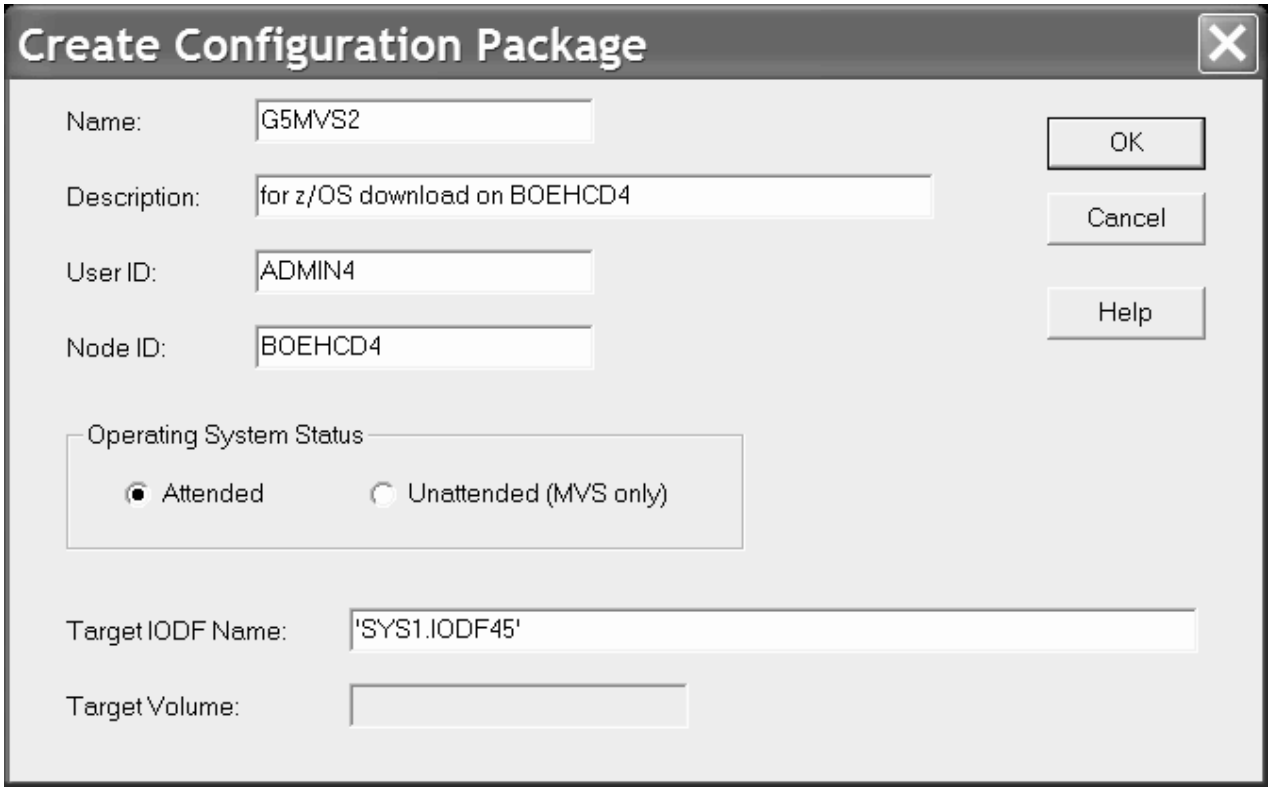


Figure 40. Create Configuration Package

z/VM:

The ***Target IODF Name*** entry field also accepts a z/VM file name. Therefore, if you specify a filetype of PRODIODF, HCM knows that you want to work with z/VM and validates the syntax against VM IODF naming conventions. This target IODF name (IODFxx) is used for building the production IODF that is sent to the target system. (xx in the IODF name stands for any two hexadecimal characters.)

As HCM allocates a new WORKIODF on the user's A-disk, an IODF with the name `IODFxx WORKIODF` must not exist, because otherwise, the package can not be built when trying to transmit it.

How to work with configuration package objects

Configuration packages contain objects, which are a selection of operating systems or processors defined in the accessed work IODF. You copy these objects as a subset from this IODF to include them into the configuration package. Before you can transmit the package with the contained objects, you must rebuild the production IODF from the master work IODF.

To fill a configuration package with objects, select the package from the list in the ***Configuration Packages*** dialog and press the ***Objects...*** button to display the ***Configuration Package Objects*** dialog. This dialog displays the objects contained in the selected package.

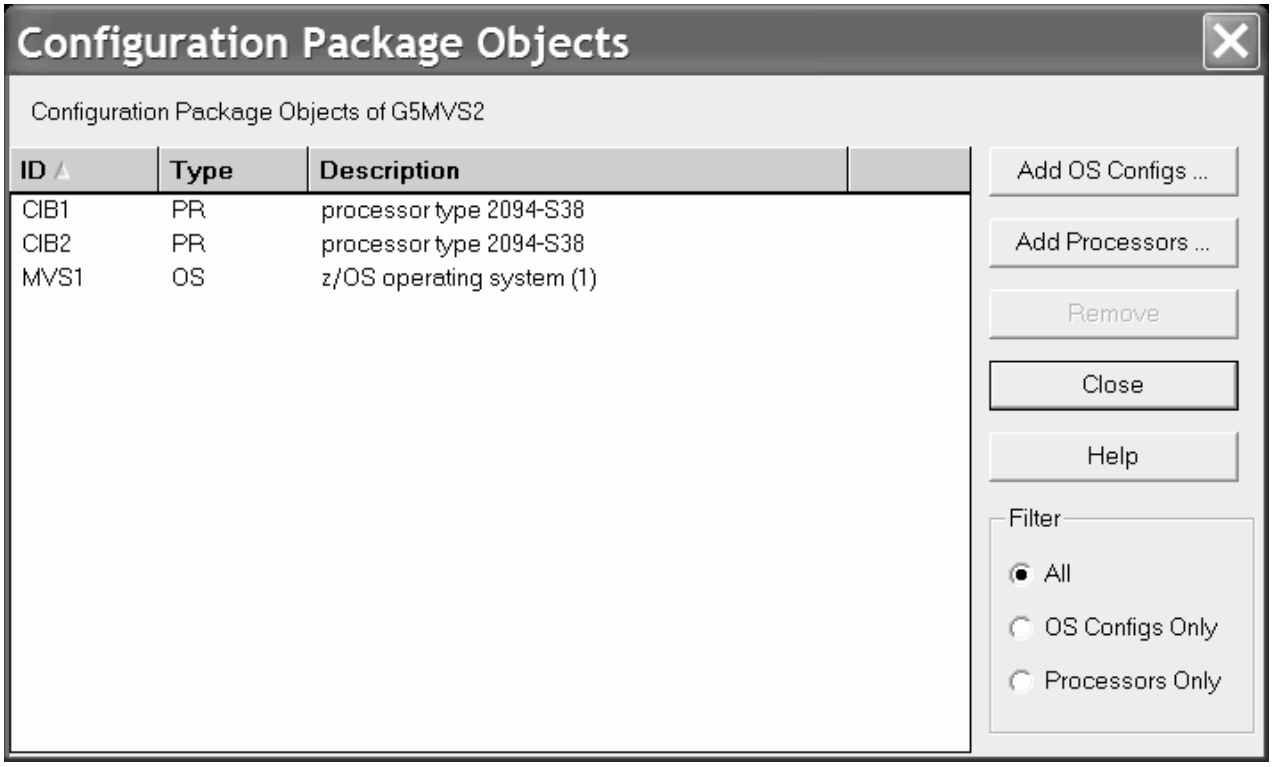


Figure 41. Configuration Package Objects

Use one of the buttons ***Add OS Configs...*** or ***Add Processors...*** to add objects of the related type to the selected package. HCM displays the appropriate dialogs which look similar. Figure 42 on page 91 is shown as an example.

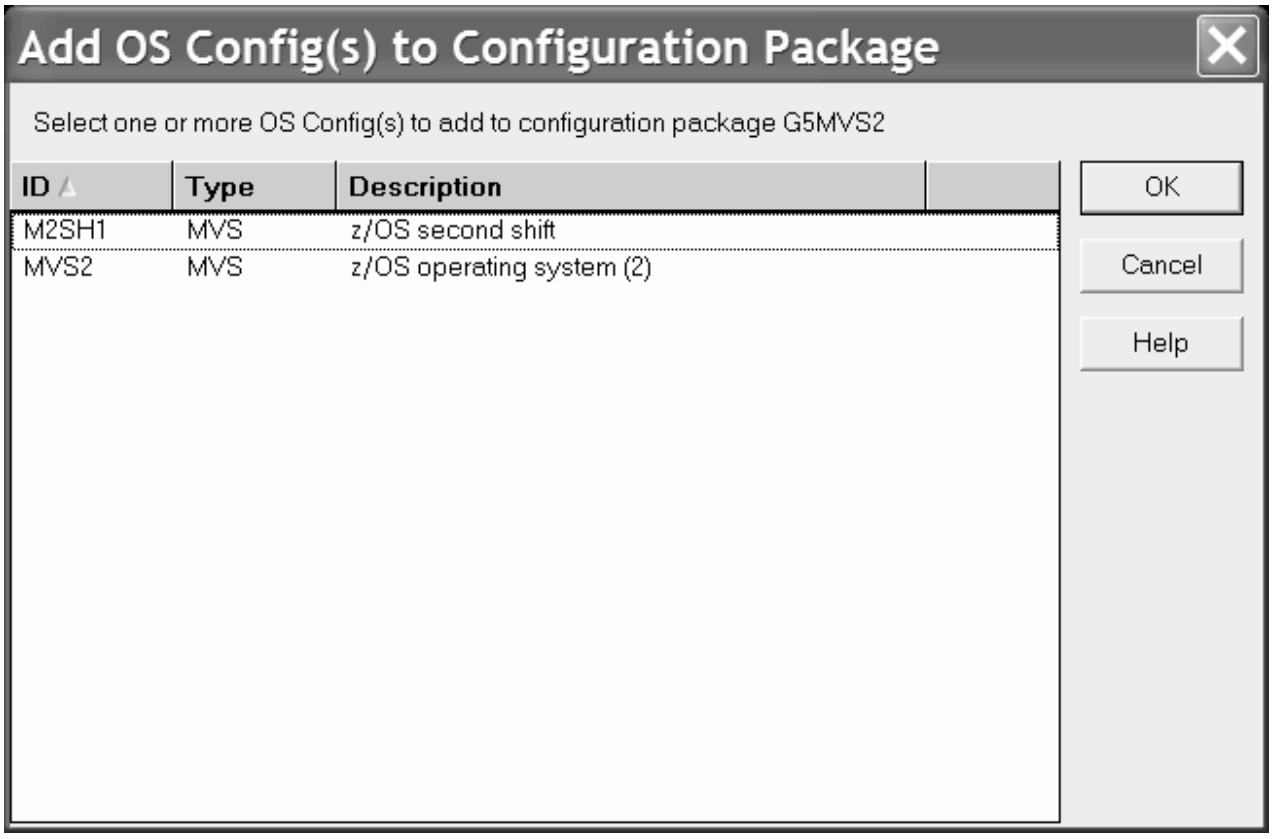


Figure 42. Add OS configuration objects

How to transmit a configuration package

Transmitting a configuration package means building a subset IODF from the objects contained in the package and sending it to the specified target system.

To transmit a configuration package, open the configuration with the associated production IODF for which the package is defined. In the ***Configuration Packages*** dialog, select the package and press the ***Transmit...*** button to display the ***Transmit Configuration Package*** dialog.

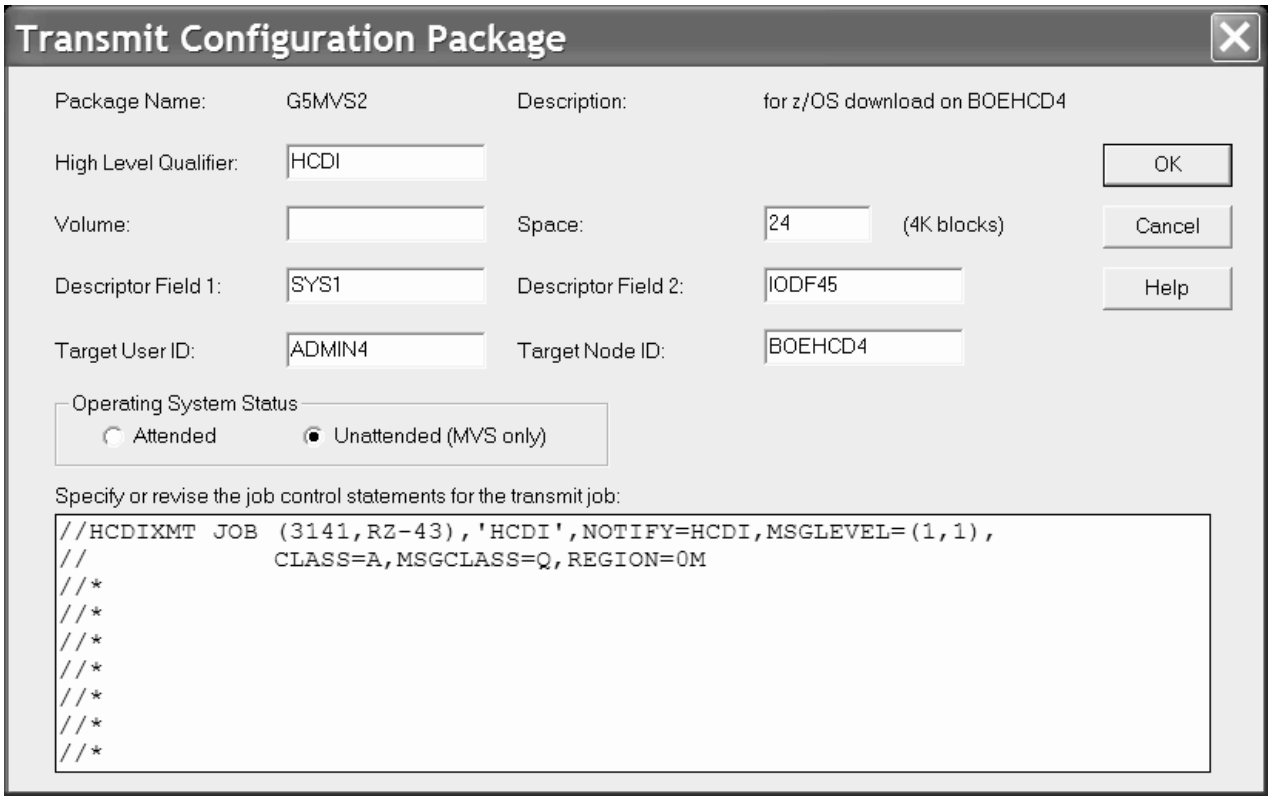


Figure 43. Transmit a configuration package

In the dialog shown in Figure 43 on page 92, you can specify and/or adapt job control statements which are used to generate the batch job for transmitting the package to the target system. This job runs on the transmitting system, where the package is defined. Changes to the statements are retained across sessions.

Specify the job card and other job control statements for the batch job. Adjust the job name, the class parameters, the user ID to be notified, and the region size to your needs.

Also, the data in the entry fields above the specified JCL is used to generate the batch job:

- The ***High Level Qualifier*** is used for creating the temporary IODFs and defaults to the HCM user ID.
- The ***Descriptor Field 1*** defaults to the HLQ and the ***Descriptor Field 2*** defaults to the second qualifier of the target IODF name in the configuration package.

Use the ***Volume*** field to specify where to save the temporary IODFs before transmitting them to the target system (required on non-SMS managed systems). This field is only applicable when sending to an unattended MVS system. Otherwise it is ignored. In the ***Space*** field, HCM proposes an estimated average size for IODF space allocation. You can modify this value.

The submitted batch job:

- creates a temporary work IODF, which receives the subset of OS and processor configuration objects defined in the package and all the related data
- creates a temporary production IODF from the work IODF

- and sends the production IODF either to an attended or unattended target system.

z/VM:

The target IODF is built on the z/VM user's A-disk by:

- allocating a new WORKIODF, using the target IODF file name (IODFxx)
- exporting and migrating the I/O configuration statements for the processors, that are included in the transmitted configuration package, into the work IODF (including the affected switch definitions)
- exporting and migrating the I/O configuration statements for the operating systems, that are included in the transmitted configuration package, into the work IODF
- building a production IODF
- transmitting the production IODF to the target system

The used work files are removed from the A-disk after a successful transmission. If any of the above steps fails, the work files remain on the A-disk for failure diagnostic and must be manually deleted before a new package transmission with the same IODF file name can be performed.

How to send a package to an unattended node

If you have specified to send the selected package to a target system running in unattended mode, use the ***Specify Import JCL*** dialog (Figure 44 on page 93) to provide appropriate job card parameters and job control language (JCL) for a job that will run on that unattended system to import the package into the desired target IODF. Modify the job statement to your needs, for example, specify the correct JES routing statement.

Also, on the receiving unattended node, you need to ensure that user authentication is correctly set up. How to do this, is described in detail in the *z/OS HCD User's Guide*.

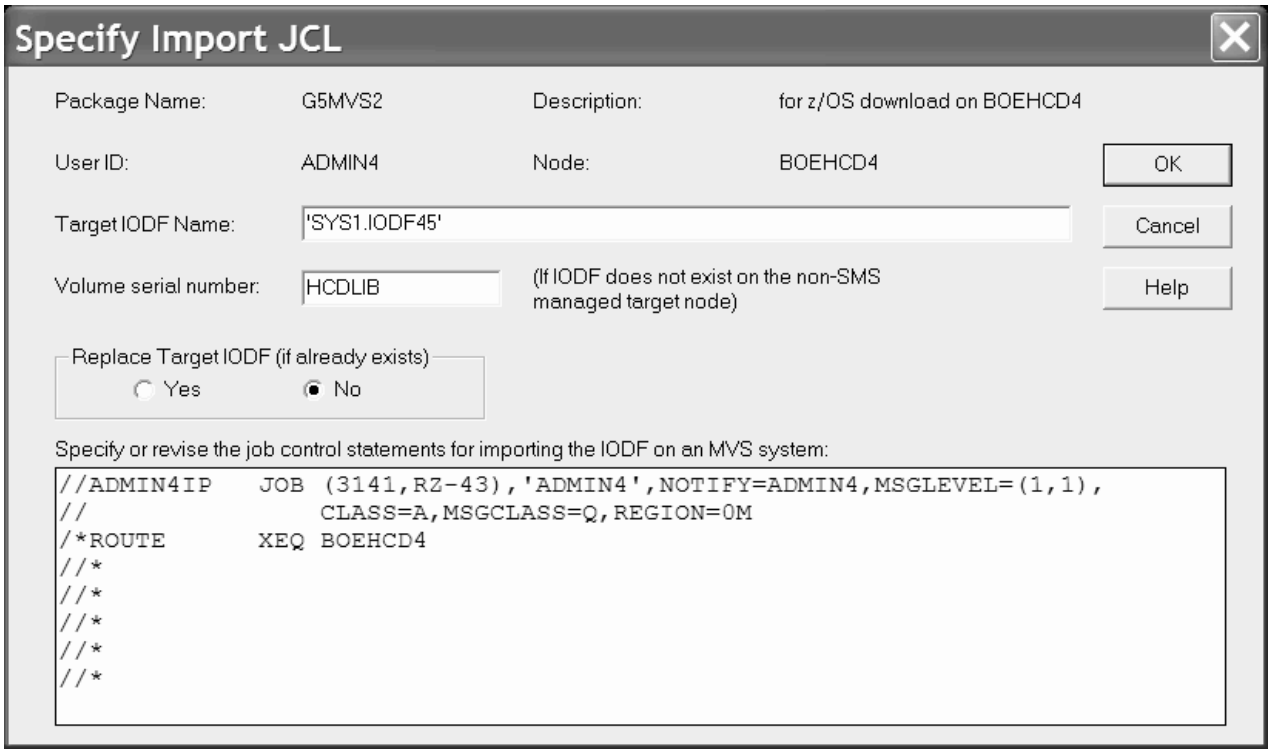


Figure 44. Specify Import JCL

Chapter 5. Stand-alone mode

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

In stand-alone 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.

Note: If in stand-alone mode, you apply physical changes to a local copy of an MCF-enabled configuration, HCM cannot synchronize with the host-based MCF immediately, but sets a so-called ***LC flag***, indicating that the configuration was locally changed. Synchronizing then is processed with the next reconnection.

Invoking the stand-alone mode

To invoke the stand-alone mode select the **Standalone** button on the **Login** dialog (see Figure 4 on page 25). As a prerequisite, your configuration file must reside on a resource on the workstation (or accessible from the workstation).

Working with the stand-alone mode

The stand-alone 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 96.
- Require a resynchronization between the IODF and the configuration file.

If you work with the stand-alone mode, the most obvious deviation from the online mode are that several menu choices are disabled. The tables presented in Appendix D, “Menu bar choices summary,” on page 425 indicate, which choices are available.

Creating objects

Most operations to create objects in stand-alone 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 stand-alone mode.

The only creation operation which is allowed is the creation of physical objects, like for example, cabinets, general boxes or crossbar switches. 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.

© Copyright IBM Corp. 1995, 2021

Printing reports, diagrams, cable labels

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

Table 11. Printing of reports

Type of printing	***.hcm/hcmz file**	***.hcr/hcrz file**
Report	Partially[2]	Yes
Existing diagrams	Yes	Yes
Cable labels	Yes	Yes

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 45. 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 stand-alone 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 stand-alone 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.

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

Upgrading

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

Resynchronizing

If HCM is running in stand-alone mode the resynchronization operation is not possible.

VOLSER reconciliation

If you have updated VOLSER data in the HCM configuration in stand-alone mode, there might be a mismatch between the VOLSER data in an HCM configuration file and its corresponding IODF on the host.

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.

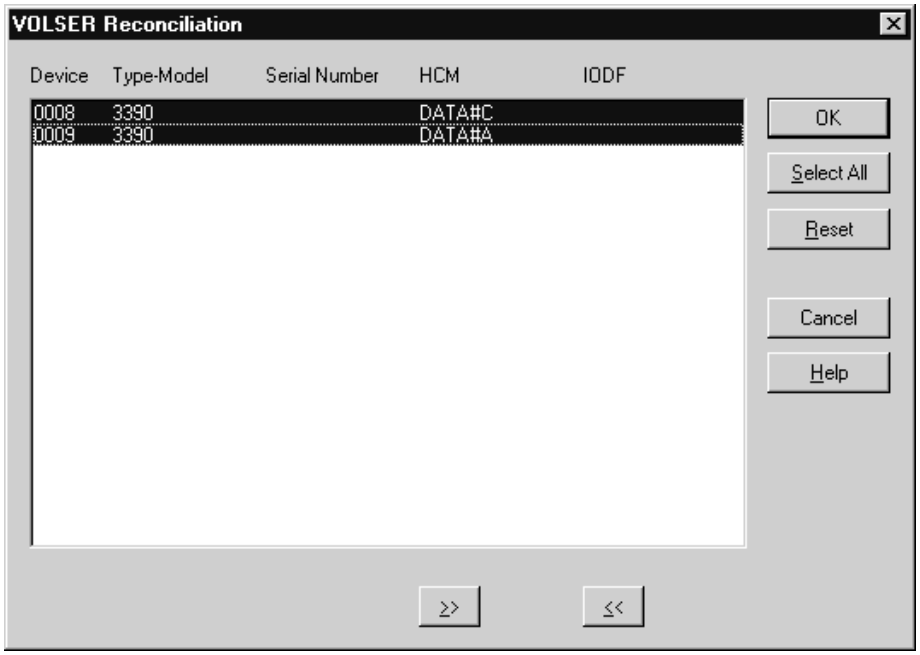


Figure 46. VOLSER Reconciliation dialog

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 PC 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.

Chapter 6. Modifying objects in your configuration

Once you have built your initial configuration, you can modify objects in the HCM configuration 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 175.

To support you in performing complex processor configuration tasks, HCM offers several copy wizards, for example, to copy an SMP processor into a channel subsystem of an XMP processor, or to copy a partition. How to use these wizards, is described in the following topics:

- “Copying a processor using a wizard” on page 109
- “Copying a channel subsystem” on page 113
- “Copying a partition” on page 116

Before using one of these wizards, you should read the general information presented in the subsequent sections:

- “General concepts of the copy wizards” on page 99
- “How HCM handles connections when using the copy wizards” on page 100

The remainder of this chapter then tells you how to create and modify all objects of an I/O configuration, starting with “Processors” on page 106 through “General boxes” on page 171.

General concepts of the copy wizards

The copy wizards support two scenarios:

- Choose the **Repeat** scenario to copy the source object to the target object without moving physical connections. Only the logical connections are copied to the target.

  The purpose of this scenario is to create a new object and keep the source object operative for further use.
- Choose **Upgrade** scenario to create a new object with physical connections moved. The wizard copies the source object to the target object, moving the physical and copying the logical connections from the source to the target. The source object remains with the logical connections only.

  The purpose of this scenario is to create a new object with full functionality and dismantle the source object. This scenario therefore helps you to consolidate your resources, for example, you may copy several SMP processors into different channel subsystems of an XMP processor.

Figure 47 on page 100 illustrates how the wizards' page flow relates to the work and data flow when using the wizards with the different scenarios:

© Copyright IBM Corp. 1995, 2021

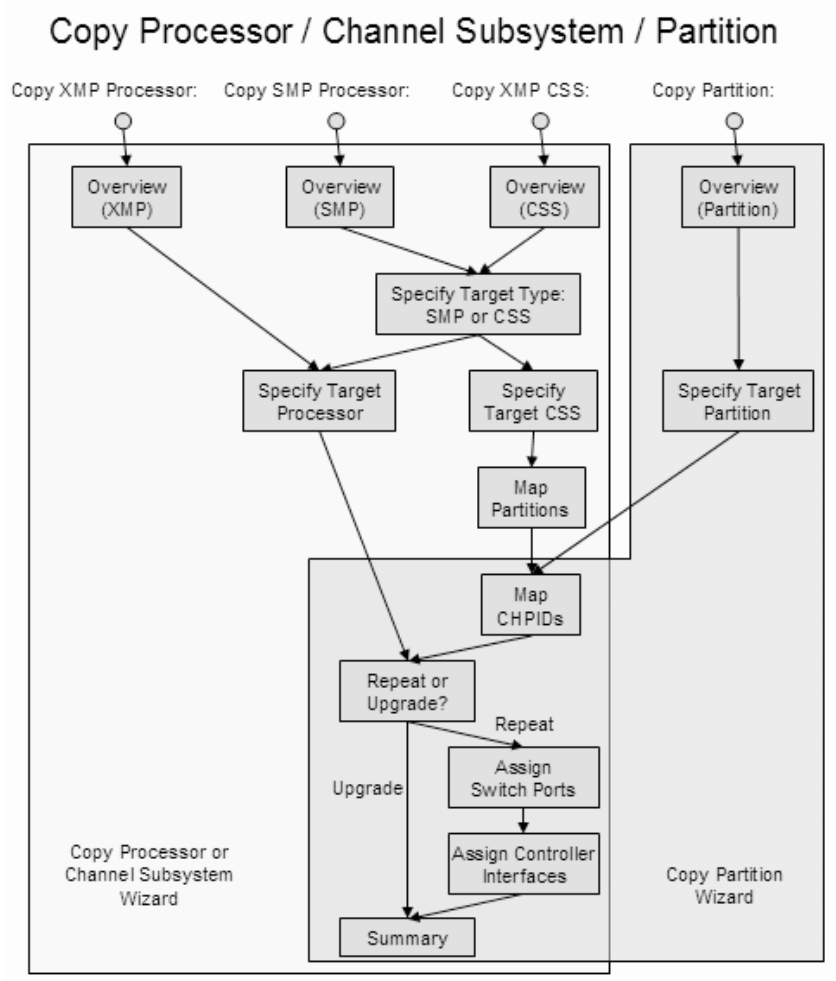


Figure 47. Wizard page flow and work flow

How HCM handles connections when using the copy wizards

When copying a processor, a channel subsystem or a partition, you must map the CHPID connections of the source object to the corresponding connections for the target processor/channel subsystem/partition. You achieve this using the wizards ' **Map CHPIDs** pages, if applicable, and - in case of the ***Repeat*** scenario - the **Assign Switch Ports** and **Assign Controller Interfaces** pages.

This chapter lists the rules how the involved copy wizards handle logical and physical connections during the copy process of source objects that contain such connections. CTC connections and CF connections are especially considered.

Mapping CHPIDs

When you copy objects, the copy wizards support you in integrating the object's connections into the context of the target. For example, when copying a partition, you can adapt the access and candidate lists of CHPIDs on the target. When copying a CSS, you can adjust spanned CHPIDs in the target. To achieve this, the copy wizards invoke the appropriate ***Map CHPIDs*** pages where required (see also Figure 47 on page 100):

- When copying a partition, CHPID mapping allows you to use the same or different existing CHPIDs on the target processor/CSS or to specify new CHPIDs that will be created by the copy operation.
- To handle spanned CHPIDs, CHPID mapping is also invoked when copying an SMP processor to a CSS of an XMP processor or a CSS of an XMP processor to a target CSS of the same or another XMP processor. You can select to have the CHPIDs spanned in the target in the same or a different way as they are spanned in the source.

Rules for CHPID mapping when CF connections are involved

CHPIDs involved in a CF connection cannot be spanned in the target, which implies that spanned CHPIDs cannot be used as a copy target for source CHPIDs involved in a CF connection. However, if you copy within a CSS, you can reuse a spanned CHPID, but you cannot change the existing spannings.

Connections in the Upgrade scenario

Remember: The Upgrade scenario creates a new object by moving the physical and copying the logical connections from the source to the target (see "General concepts of the copy wizards" on page 99).

Logical connections

Consider a logically connected CHPID which is not a CF CHPID. A CHPID might be logically connected to one or more control units, and additionally to a switch port. Any logical connection is copied from a source CHPID to the corresponding target CHPID, regardless of the type of connection. Note that logical connections are indeed **copied, not moved.** There are, however, two exceptions that apply for the Upgrade scenario, where a logical connection is not copied, but moved and where, to enable moving the connection as required, the source CHPID is disconnected from the switch port:

1. if the source CHPID is not spanned, and the source CHPID differs from the target channel path ID, and the target channel path ID is not connected
2. also , if the source CHPID equals the target CHPID, and the source processor equals the target processor, and the source CHPID is not spanned, and the target CHPID is not connected.

Moving existing physical connections

Physical connections are only moved in the Upgrade scenario but not in the Repeat scenario. Logical connections which might use the physical connection that is moved, are not considered in this section.

Using the copy wizards for handling a physical connection indeed means **moving** the physical connection from a source CHPID to a target CHPID: after a successful copy operation, the target CHPID will have the source CHPID 's physical connection, while the source CHPID no longer will have a physical connection. Logical connections remain on the source CHPID, though. For further discussion, you need to distinguish different types of physical connections, where the last connection object of the fiber link is the distinguishing attribute:

1. CHPID <---> switch port connections
2. CHPID <---> controller interface connections
3. CHPID <---> patchport/general box port connections where the patchport/general box port is not further connected to a switch port, controller interface, crossbar switch, or CHPID
4. CHPID <---> crossbar switch connections
5. CHPID <---> CHPID connections (which can only be CTC or CF connections)

Each of these connections may also contain an arbitrary number of patchports, general box ports or cables.

In case of any of the first four connection types, this physical connection is moved from the source CHPID to the corresponding target CHPID. This means that the target CHPID is being physically connected exactly the same way as the source CHPID has been connected before the copy operation. All physical objects which are part of the connection (patchports, general box ports, converters) are also moved from the source link to the target link in exactly the same order, and all cables are also moved from the source CHPID's link to the target CHPID's link. Consequently, the source CHPID no longer will be physically connected which means that if the source CHPID was involved in a logical connection before the copy operation, *'logical without physical connection'* visual warnings will appear in the configuration diagram.

With the fifth connection type, you either have a CTC connection or a CF connection. The following sections discuss the rules that apply to these connections:

- "Rules for CTC connections" on page 102

- "Rules for CF connections" on page 102

Rules for CTC connections

- If both CHPIDs of the CTC connection are located on the source object, and both source CHPIDs are mapped to target CHPIDs, HCM ensures that the CTC connections are completed for the target object. HCM also ensures that CUADD values are also correctly set (if such a change is needed). This way, it is ensured that both physical and logical connections are established correctly on the target.

  This is also true if you have an FCTC connection over a single FC CHPID.
- If both CHPIDs of the CTC connection are located on the source object, and if just one of the source CHPIDs is mapped to a target CHPID, then a CTC connection is established between the new target CHPID and the unmapped source CHPID.

  The physical part of the link on the source CHPIDs is removed from the source CTC connection and is connected to the new target CHPID.

  As the logical connection between the source CHPID and the CTC control unit(s) and CTC device(s) is not disconnected, the **Unused...** button will be available on the **CTC Connections** dialog. For details on unused CTC control units, devices and connections to CHPIDs, see "Editing a CTC connection" on page 197.

  The **Unused...** button will also be available if just one CHPID of the CTC connection is located on the source object and is mapped to a CHPID on the target object, and if the connected destination CHPID is located on a different processor or channel subsystem. That is, HCM establishes the new CTC connection between the mapped CHPID on the target processor and the unmapped destination CHPID. The reason is as before, that the logical connection between the mapped source CHPID and CTC control unit(s) and CTC device(s) is not disconnected.

After successful copy operations, you should verify your CTC connections via the ***CTC Connections*** dialog and check whether some unused CTC control units, CTC devices or CTC CHPIDs are reported.

Rules for CF connections

Moving a CF connection to a target object (processors, CSSs or partitions) must not influence objects on the target. Therefore, in certain cases, CF connections cannot be copied because this operation would affect other connections.

If one of the CF connection CHPIDs is not copied (deselected in the ***Map CHPIDs*** page), then the CF connection is not copied and the physical connectivity is also kept on the source CF connection.

The following rules apply for the different object types:

- Copying a processor
  - All CF connections which are not deselected in the ***Map CHPIDs*** page are copied to the new target processor.
  - For CF connections within the processor, both source and target of the connection are moved to the new processor.
  - If the CF connection to be copied is completely within the same processor, the source connection is kept in the source processor. Otherwise, if both ends of the connections are in different processors, the origin of the connection is moved to the target and disconnected in the source.
- Copying a CSS
  - All CF connections are copied to the new target CSS, with the following exceptions:
    - Connections established with spanned CHPIDs in the source CSS are not copied.
    - If a CSS with internal CF connections is copied to another processor, both connection ends must be within the same CSS. Otherwise, the CF connections are not copied.
  - If both connection ends of the source CF connection are within the source CSS, the source connection is kept. Otherwise, if the connection target is in a different CSS, the source connection is moved to the target and disconnected in the source.

- Copying a partition
  - All CF connections using dedicated CHPIDs are moved to the target partition.
  - If a source OS partition that you want to copy within the same CSS is involved in CF connections, then the following applies:

    A CF connection using a spanned, shared or reconfigurable CHPID is copied if the CHPID is not mapped to a different CHPID in the target. Shared or spanned source CHPIDs must not be spanned to different CSSs in the target (then the same physical cable is used and the copy operation just enhances the access list).
  - A CF connection using a spanned, shared or reconfigurable CHPID is not copied if the target partition is in a different processor or CSS. A CF connection is also not copied if the copied partition is a CF or CF/OS partition.
  - Internal CF connections are not copied if the partition is copied to a different processor.
  - If the partition is copied within the same CSS, the source connection is kept, except for a dedicated CHPID. Otherwise, if the partition is copied to a different processor/CSS, the source connection is moved to the target and disconnected in the source.

Remarks on converters

If the interface type of the source and the target CHPID coincide, you need not care about converters. If a converter has been part of the source CHPID 's link, the link of the target CHPID will also contain a converter. If this converter is labelled , then it will become part of the target CHPID's link. Otherwise, a generic converter will be added .

If the interface type of the source CHPID differs from that one of the target CHPID, then a converter might either be removed or added to the configuration. If a generic converter was part of the source link, it is just removed from the configuration. Otherwise, in case of a labelled converter, this converter still exists in the configuration, but it is no longer used in any link. You can use this labelled converter to replace, for example, another generic converter. If a new converter needs to be added into the target link, a generic converter will be created . This is also true if labelled , but currently unused converters exist in the configuration.

Exception

No physical connection is being copied:

- if the source CHPID gets mapped to a target CHPID in the following scenarios:
  - copy SMP processor to channel subsystem
  - copy channel subsystem to channel subsystem
  - copy partition
- and if at the same time the target CHPID is physically connected

Note: If the target CHPID to which the source CHPID is mapped, is not the lowest instance of a spanned CHPID definition, but the lowest CHPID instance is already physically connected, then the selected target CHPID is also considered to be physically connected.

Remarks on visual warnings

'Logical without physical connection' visual warnings can appear in the diagram after you completed an Upgrade scenario, because both the source and the target CHPID will be logically connected to the same connection target - if none of the mentioned exceptions applied - for example in the following cases:

- CHPID <---> switch port connection: if the physical connection has been moved, the switch port no longer has a physical connection to the source CHPID and will be flagged with this warning.
- CHPID <---> controller interface connection: if the physical connection has been moved, the controller interface no longer has a physical connection to the source CHPID. The affected controller segment will be flagged with this warning.

Connections in the Repeat scenario

Remember: The Repeat scenario copies the source object to the target object without physical connections. Only the logical connections, but no physical connections, are copied to the target (see “General concepts of the copy wizards” on page 99).

Logical connections in the Repeat scenario

As in “Logical connections” on page 101, consider a logically connected CHPID which is not a CF CHPID. As in the Upgrade scenario, any logical connection is copied from a source CHPID to the corresponding target CHPID, regardless of the type of connection. The subsequent sections handle in detail CTC connections and also CF connections. Once more, note that logical connections are indeed **copied, not moved.**

Assigning new connections

New physical connections are only established in the Repeat scenario. You achieve this via the **Assign Switch Ports** and **Assign Controller Interfaces** wizard pages. Existing physical connections are only copied in the Upgrade scenario, but no new physical connection is assigned in such a scenario. Also, any logical connections are not considered in this section which might use the physical connection that is assigned. Emphasis is on CTC connections and CF connections.

As only new physical connections are established in this scenario, and as no existing physical connections are copied, the link (physical connection between the target CHPID and its connection targets) will not contain any patchport , general box port, or cable. If required, you must define all such physical configuration objects manually to the link after the copy operation succeeded.

As mentioned above , there are two wizard pages which help you establish the new physical connections for the target CHPIDs. The **Assign Switch Ports** page only lists those target CHPIDs for which the corresponding source CHPID is connected to a switch port. For all shown target CHPIDs, you can select a switch port as connection target in order to establish a complete physical path from the target CHPID onto controllers.

On the **Assign Controller Interfaces** wizard page, you can assign controller adapter interfaces to target CHPIDs. Only those target CHPIDs are listed on this wizard page for which the corresponding source CHPID is connected to a controller interface. The target CHPID must be connected to the same controller as the corresponding source CHPID. If not enough labelled controller adapter interfaces are available on a controller, HCM will provide unlabeled controller adapters and unlabeled controller adapter interfaces which are indicated by ’INT’. You can label those adapters and their interfaces yourself afterwards, see “Editing labels of adapters and interfaces” on page 136.

Several CHPID types are usually connected to specific control unit types. Some CHPID <---> controller connections cannot be selected on the **Assign Controller Interfaces** page, but are automatically established by HCM. Connections to which this applies involve a controller with either an OSA type control unit, an IQD control unit or an FCP control unit.

When connecting a mapped target CHPID to a controller interface, the interface type of the controller interface may differ from the interface type to which the source CHPID is connected. In such a case, HCM adds a generic converter to the configuration or removes the previously needed converter.

Note that a mapped target CHPID will be neither shown in the **Assign Switch Ports** page nor in the **Assign Controller Interfaces** page and therefore will have no physical connection at all in the following cases:

- A source CHPID is just connected to a patchport or a general box port, but not connected to a switch port or to a controller adapter interface.
- A source CHPID is connected to a crossbar switch, but not connected to a switch port or to a controller adapter interface.

If , however, a CHPID is connected to a crossbar switch, and furthermore is connected to either a switch port or a controller adapter interface, then this CHPID is shown on either the **Assign Switch Ports** page or on the **Assign Controller Interfaces** page. The logical connection is copied and the physical connection will be established between the CHPID and the selected switch port or controller adapter interface.

Rules for CTC connections

You need to distinguish between switched CTC connections and point-to-point CTC connections. For FCTC connections, we have a two-digit CUADD value where the first digit indicates the CSS ID for XMP processors and the second digit indicates the MIF ID. If a channel subsystem is involved in the copy operation for a FCTC connection, the CUADD value might need to be considered as well.

Note that all CTC control units and devices which are connected to any source CHPID are logically connected to the corresponding target CHPID, but the CUADD values are not updated so that the CTC connections on the target CHPID might not be complete.

1. Switched CTC connections: no new CTC connection is established.
   a. Case 1: Only one of the involved CTC CHPIDs is defined to the copied source object.
      i) If the source CHPID is not mapped to a target CHPID, a new CTC connection cannot be established anyway.
      ii) If the source CHPID is mapped to a target CHPID, then the target CHPID is connected to a different switch port than the CTC control units to which a logical connection is established. Hence, as the link address is not correct, no new CTC connection can be established, neither for SCTC nor for FCTC connections.
   b. Case 2: Both of the involved CTC CHPIDs are defined to the copied source object.
      i) If not both CHPIDs are mapped to a corresponding target CHPID, a new CTC connection cannot be established even though the logical CHPID <---> control unit connection is established. If one CTC CHPID is mapped to a target CHPID, the same situation as for scenario “1.a.ii” on page 105) applies.
      ii) If both involved CTC connections are mapped to a target CHPID, the logical connection between the CTC control units and the CTC CHPIDs is established, but the link addresses of the CTC control units do not coincide with the ports on the switch(es) to which the CTC CHPIDs are connected (see also previous scenarios) . Hence, the CTC control unit definitions are not correct for a complete CTC connection.
2. Point-to-Point CTC connections: As no physical connection is copied or moved, the source CHPID(s) is (are) physically still connected to the connection target. As also the logical connections between the source objects are not modified, all source CTC connections remain unchanged.
   a. Case 1: Only one of the involved CTC CHPIDs is defined to the copied source object
      i) If the source CHPID is not mapped to a target CHPID, a new CTC connection cannot be established anyway.
      ii) As this CHPID has no connection target, no new point-to-point CTC connection can be established. This applies to both the SCTC and the FCTC case.
   b. Case 2: Both of the involved CTC CHPIDs are defined to the copied source object
      i) If not both CHPIDs are mapped to a corresponding target CHPID, a new CTC connection cannot be established as either no CHPID is mapped anyway, or a connection target is missing for the other mapped target CHPID.
      ii) If both involved CTC connections are mapped to a target CHPID, a new CTC connection is established in any case for an ESCON point-to-point CTC connection, as both the logical and the physical connections are established. In case of FICON point-to-point CTC connections, a CTC connection is established as long as the two-digit CUADD value for the source as well as the target CTC connection are correct. If the CUADD value needed for the mapped target CHPIDs differs from that of the source CHPIDs, no new FCTC connection is established.

After the copy operation succeeded, you should verify your CTC connections via the ***CTC Connections*** dialog and check whether some unused CTC control units, CTC devices or CTC CHPIDs are reported.

Rules for CF connections

The logical and physical connections between source CF CHPIDs and their connected target CF CHPIDs remain unchanged on the source. Copying them to the target (both physically and logically) works as follows:

- Copying a processor:
  - CF connections within the source processor (either with internal or external CF CHPIDs) are also established at the target processor with new CF control units and devices. Otherwise, CF connections that connect to a different processor are not established at the copy target processor/CSS.
- Copying a CSS:
  - If the CF connection is within the source CSS (either with internal or external CF CHPIDs), a new connection will be established in the target processor with new CF control units and devices . If the CF connection is across CSS boundaries (which includes connections with spanned CHPIDs), no connection is established at the target processor.
- Copying a partition:
  - CF connections are not copied; that is, no new CF connection is established at the target.
  - An OS partition is included in the access list of a matching shared or spanned CF CHPID that may already have a CF connection at the copy target (that is, if a CF connection already exists between the two target CHPIDs).
  - A CF or CF/OS partition is not included in the access list of a matching shared or spanned CF CHPID. (If a dedicated CF CHPID does not yet exist in the CSS of the target, it is copied without establishing a CF connection.)

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 an 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, or the processor details 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 A, “Alternate processor definitions,” on page 415 >.>

There are two versions of the **Processor** dialog, one for SMP processors(Figure 48 on page 107) and one for XMP processors (Figure 49 on page 108). The one for XMP processors contains a **Channel Subsystems** section which lists the channel subsystems (CSS) defined for the processor. For each

channel subsystem, you can see the maximum number of devices that can be defined in each available subchannel set and the actual number of defined devices.

You can tailor the information that you want to be displayed in the **Channel Subsystems** section. Pressing the **Columns...** button lets you add or remove columns to or from the **Channel Subsystems** box.

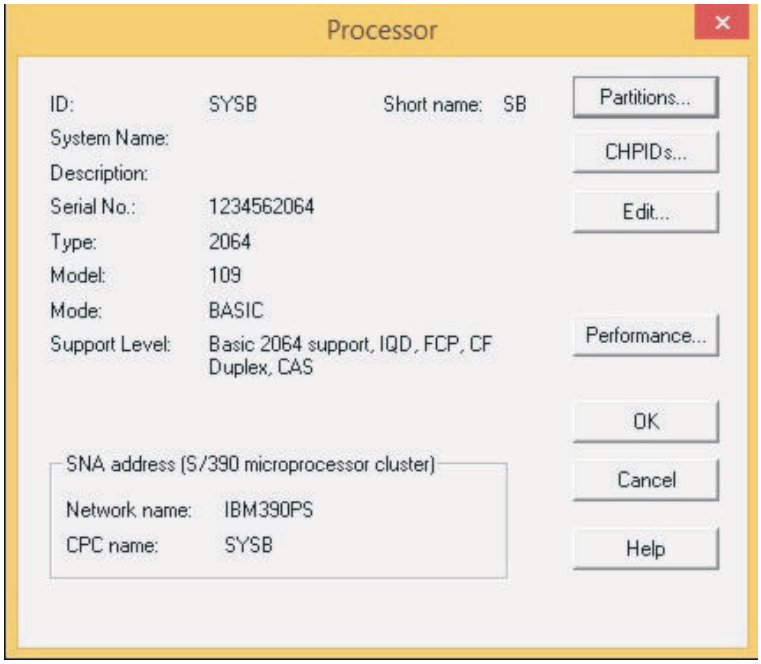


Figure 48. Processor dialog for SMP processors

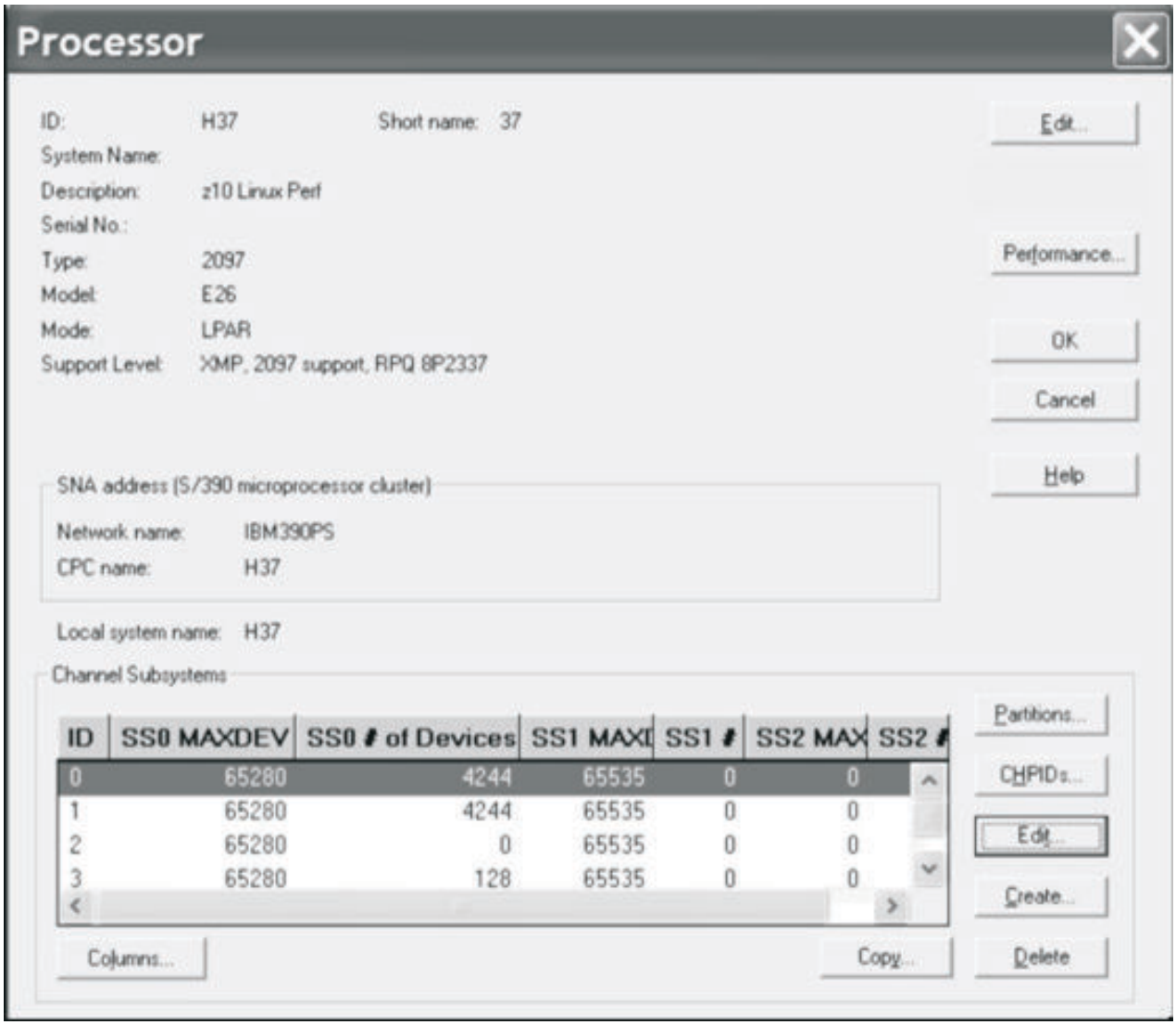


Figure 49. Processor dialog for XMP processors

To edit partitions and channels for XMP processors, you must select a CSS first, as these objects are pertaining to a specific CSS.

Use the **Copy...** button in the **Channel Subsystems** section to open the **Copy Channel Subsystem** wizard. This wizard lets you copy a channel subsystem of an XMP processor into the same or another existing XMP processor or into a new SMP processor. For more information refer to "Copying a channel subsystem" on page 113.

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 processor. 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 “Processors” on page 106).

Note that the **Short name** of a processor is part of the physical machine.

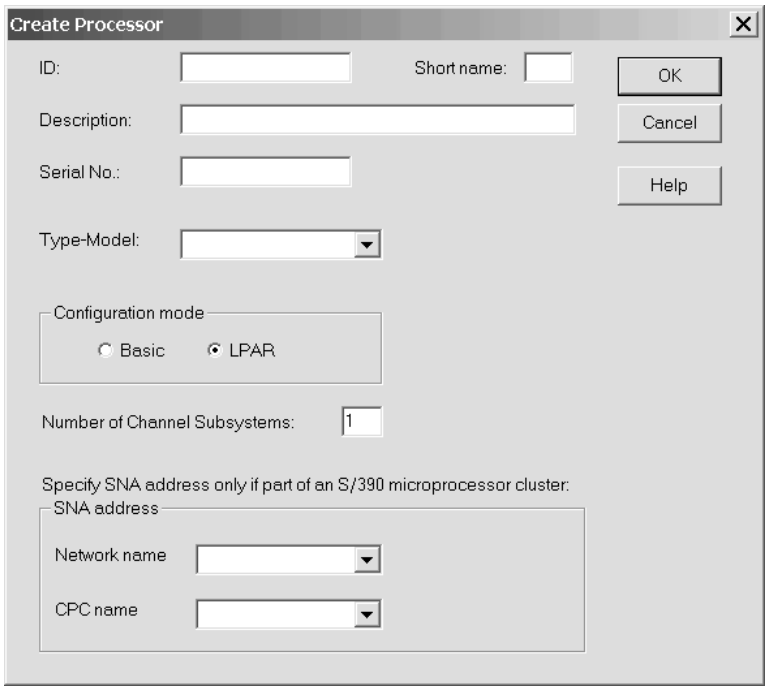


Figure 50. Create Processor

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 415 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). If you are unsure about a support level's properties, you can press the **Details** button to receive an explanation of the used terms and abbreviations.

Copying a processor using a wizard

Two wizards are available for copying processors:

- **Copy XMP Processor**: Invoke this wizard from the context menu of a selected XMP processor or from the **Edit a Processor** dialog.
- **Copy SMP Processor**: Invoke this wizard from the context menu of a selected SMP processor or from the **Edit a Processor** dialog. It lets you copy this processor into a new SMP processor or into an XMP processors channel subsystem. As the functionality for copying an SMP processor into a CSS is basically the same as just copying a channel subsystem, see “Copying a channel subsystem” on page 113 for information about how to use the **Copy Channel Subsystem** wizard.

These Copy Processor wizards help you to create a copy of an existing XMP processor or SMP processor within the same IODF / HCM configuration file. The target processor must not yet exist.

This wizard helps you to perform the following steps:

1. Specify a target processor for the copy request.
2. Choose either the **Repeat** or the **Upgrade** scenario. If you choose the **Repeat** scenario, then no physical connection of CHPIDs on the source processor are moved. If you select the **Upgrade** scenario, the physical connections of the CHPIDs on the source processor are moved to the target CHPIDs (if possible). Check the wizard page and the corresponding help page for more detailed information about the two scenarios.
3. Assign switch port connections for all target CHPIDs whose corresponding source CHPIDs have a switch port connection. Target channel path IDs with existing switch port connections are not shown in this step. This step is only invoked if you select the **Repeat** scenario in step "2" on page 110.
4. Assign controller interface connections for all target CHPIDs whose corresponding source CHPIDs are physically connected to a controller. Target channel path IDs with existing physical connections are not shown in this step. This step is only invoked if you select the **Repeat** scenario in step "2" on page 110.
5. Examine the summary of the processor information that is to be copied. Click **Finish** to perform the operation.

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 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 the buttons to invoke special utilities:

PCIe...
This button invokes the ***PCIe Functions*** dialog. Use this dialog to create, modify, or delete PCIe functions of the selected processor. See "Managing PCIe functions" on page 111 for further information.

CMT...
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 "Using the CHPID Mapping Tool Support utility" on page 73 for further information.

WPT...
This button invokes the **WWPN Prediction Tool Support** dialog. Use this dialog to export the FCP channel/device specific data of the I/O configuration from the currently accessed IODF into a comma-separated value (CSV) output format. The generated file can be used as input to the WWPN Prediction Tool to assign world-wide port names to virtualized FCP ports for configuring SAN devices. See "Using the WWPN Prediction Tool Support" on page 242 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 29.

Deleting a processor

Select the processor object in the configuration diagram and press the ***Delete*** key on your keyboard. HCM displays 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 displays 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 A, “Alternate processor definitions,” on page 415.>

Managing PCIe functions

To define, delete or change PCIe functions, click on the **Edit...** button in the **Processor** dialog to display the **Edit Processor** dialog. Then click on the **PCIe...** button to open the **PCIe Functions** dialog (see Figure 51 on page 111).

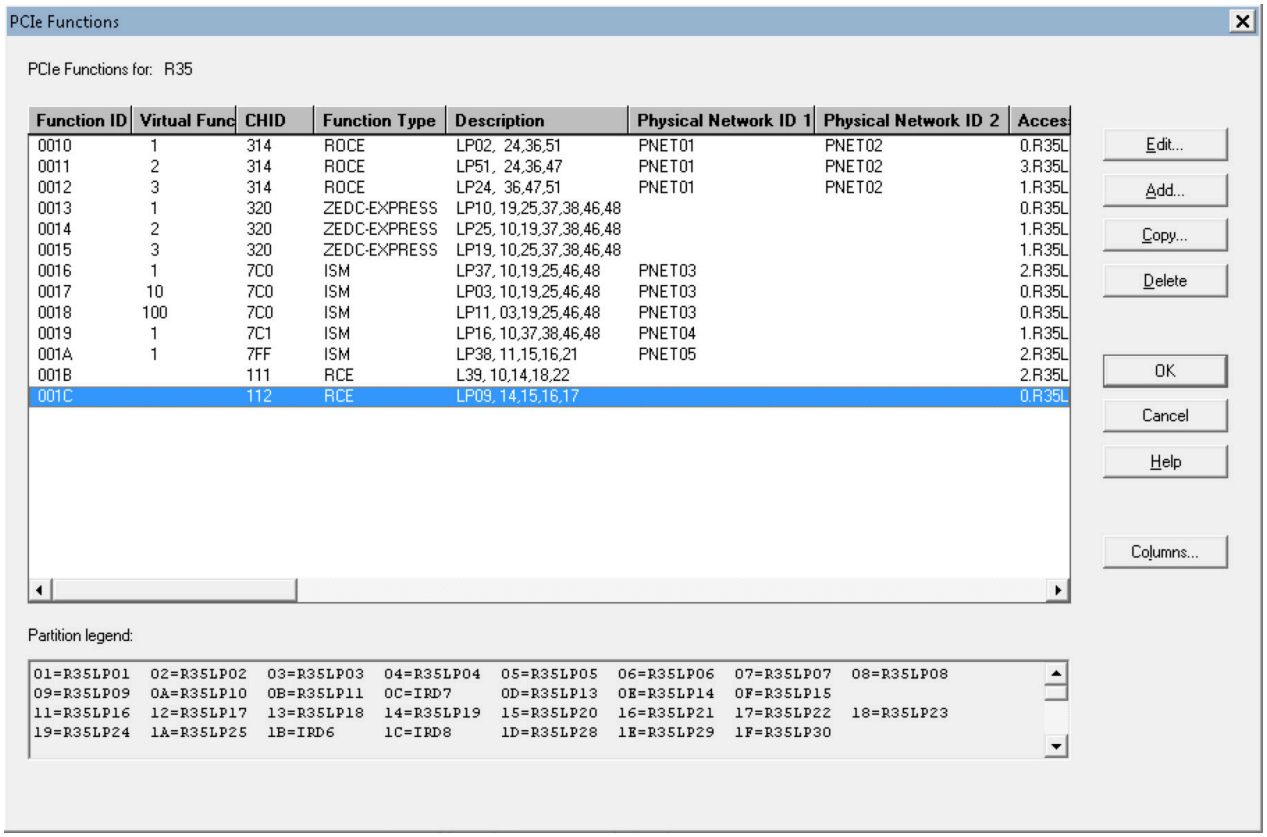


Figure 51. PCIe Functions dialog

The **PCIe Functions** dialog shows the existing PCIe functions defined in the currently accessed IODF. Each PCIe function specifies a function type and a channel identifier (CHID). Multiple PCIe functions may be specified to the same CHID value provided that each of these functions defines a unique virtual function (VF) number.

For defined PCIe functions, the **PCIe Functions dialog** from Figure 51 on page 111 also shows an optional description, and the partitions assigned in their access or candidate lists, shown explicitly by name and also as overview. You can also include columns into the dialog that show the ***Physical Network IDs*** (not visible in Figure 51 on page 111).

This dialog offers all required actions to manage PCIe functions in your configuration. Select any of the listed PCIe functions to either edit, copy or delete it, using the respective button.

To define new PCIe functions, you can use either the **Add...** or **Copy...** button.

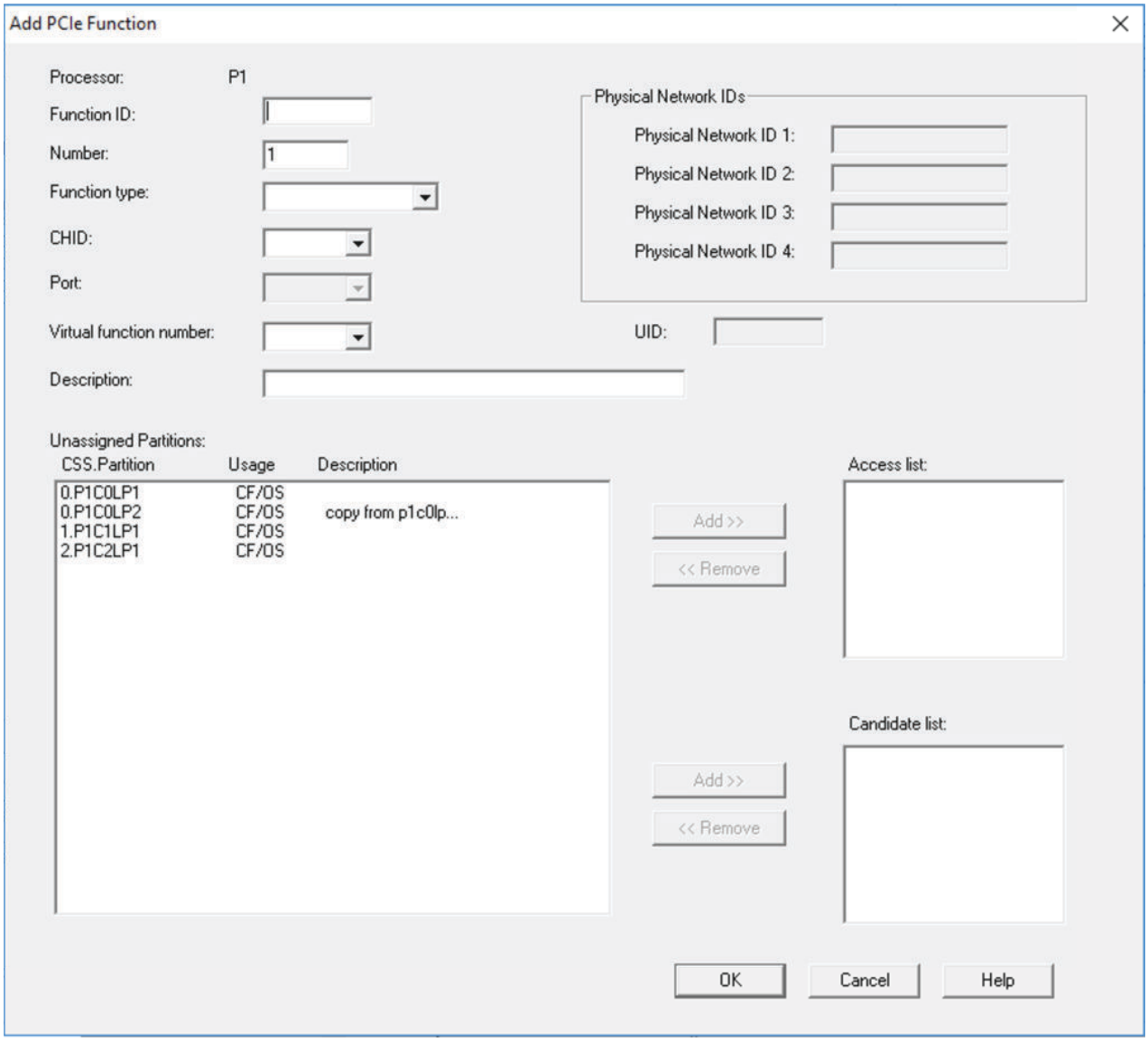


Figure 52. Add PCIe Function dialog

You specify a new PCIe function by its required ***Function ID***, ***Function type*** (for example: ROCE), and a CHID value. These attributes describe the related PCIe adapter card in the assigned slot of the I/O drawer. The description is optional. ***The Virtual function number*** is supported for specific function types. Multiple PCIe functions may be defined for the same CHID by assigning a unique virtual function number to each of these functions. The ***UID*** is supported for the appropriate PCIe functions. If any partitions with **UID** uniqueness checking flag YES are in the ***Access list*** or ***Candidate list***, the ***UID*** value has to be unique in the scope of the partitions.

HCM supports creating multiple-PCIe functions in one step by specifying the ***number.***

At least one partition must be assigned in either the access or candidate list.

HCM also supports overgenned PCIe functions. You define overgenned PCIe functions by providing an asterisk (*) for the CHID value. Overgenned functions are validated like other PCIe functions but they are excluded from the IOCP input statements and from any dynamic activation.

You can edit any properties of a PCIe function later, except for the ***Function ID***.

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.

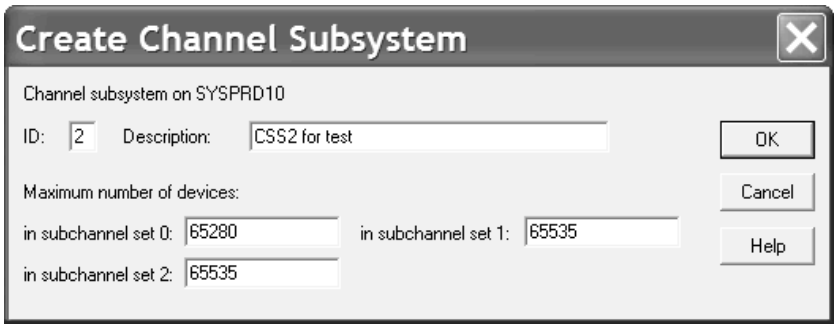


Figure 53. Create Channel Subsystem

If you want to edit an existing CSS, select it and press the **Edit...** button in the Channel Subsystems section. The upcoming **Edit Channel Subsystem** dialog lets you change the maximum number of devices for all supported subchannel sets 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 context menu offers you choices to edit the channel subsystem or to work with its CHPIDs and partitions.

Copying a channel subsystem

HCM offers the **Copy Channel Subsystem** wizard that supports you to copy a channel subsystem:

- copy a channel subsystem into the same or another existing XMP processor
- copy a channel subsystem into a new SMP processor

Invoke this wizard by pressing the **Copy...** button in the **Channel Subsystems** section of the **Processor** dialog for XMP processors(Figure 49 on page 108).

The wizard guides you through the following steps.

1. Choose the target channel subsystem or SMP processor.
   a. If you have chosen to copy the channel subsystem into a new SMP processor, the wizard offers a dialog which is similar to the **Create Processor** dialog. Then it proceeds to step “4” on page 114.
   b. If you have chosen to copy the channel subsystem into a new target channel subsystem, specify the attributes of the target CSS. The wizard continues with step “2” on page 114.

2. Assign partition names in the target CSS for all partitions which are copied from the source CSS. The partition names of the source CSS are used as the default for the target CSS, if these names will be unique also within the target XMP processor . For duplicate names, you need to specify new names for the target CSS.
3. Map CHPIDs of the source CSS to CHPIDs in the target. CHPIDs that can be defined as 'spanned', can be spanned over existing CHPIDs in other channel subsystems of the target processor.
4. Choose how you want to handle the connections:
   - If you want to copy the logical connections without moving the physical connections, choose the **Repeat** scenario.
   - If you want to copy the logical connections and move the physical ones from the source to the target, choose the **Upgrade** scenario.

   Note: For a better understanding of the **Repeat** and **Upgrade** scenarios and the wizard 's panel flow depending from your selection, refer to “General concepts of the copy wizards” on page 99.
5. If you select the **Upgrade** scenario, the physical connections of the CHPIDs on the source CSS are moved to the target CHPIDs, if possible.

   If you selected the **Repeat** scenario, then the wizard helps you to perform the following steps:

   a. Assign switch port connections for all target CHPIDs whose corresponding source CHPIDs have a switch port connection. Target CHPIDs with existing switch port connections are not shown in this step.

   b. Assign controller interface connections for all target CHPIDs whose corresponding source CHPIDs are physically connected to a controller. Target CHPIDs with existing physical connections are not shown in this step.
6. View the messages at the summary page, verify the results and click the **Finish** button.

The wizard displays appropriate dialogs as you proceed through these steps. For further information about how to complete these dialogs, refer to the online help that is available for each wizard page. Also refer to “How HCM handles connections when using the copy wizards” on page 100.

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 displays 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.

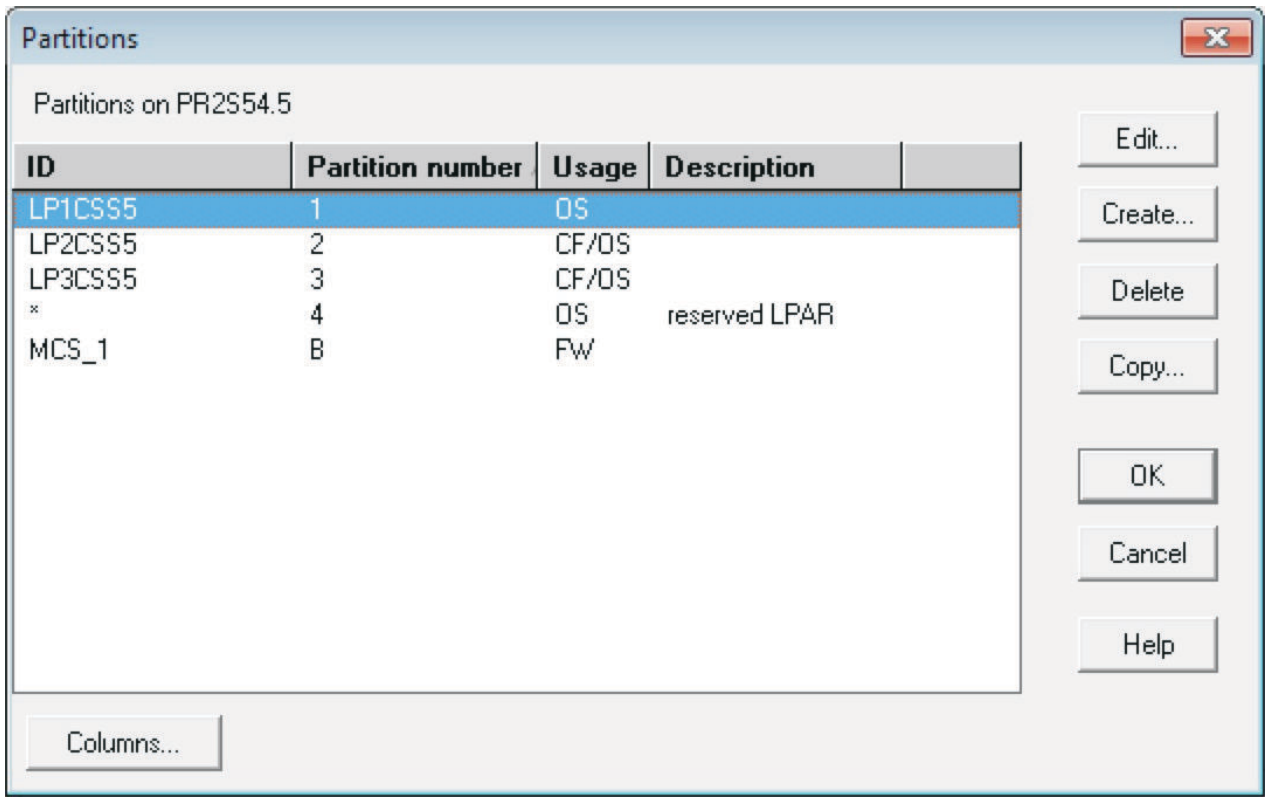


Figure 54. Partitions

Creating a partition

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 appears in the **Partitions** dialog list box. It also appears 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). Or the partition will only be used internally by firmware when the usage type is FW.

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 126 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 Partition** 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.

Copying a partition

HCM offers a **Copy Partition** wizard to let you copy a source partition configuration to a target partition. You can create the target partition on the same processor as the source partition, or on a different processor or channel subsystem. Reserved partitions can be neither source nor target of the copy action.

You can invoke this wizard with the **Copy...** button for a selected partition from the **Partitions** dialog, or from the **Edit a Partition** list. Also, you can invoke it from the **Copy...** menu item of a partition 's context menu .

This wizard helps you to perform the following steps. For a better understanding of the wizard 's panel flow, refer to Figure 47 on page 100.

1. Specify the target partition. The wizard dialog for this step queries all information required for the target partition, for example, its target processor and channel subsystem, if the target is an XMP processor, the partition ID and number. You cannot change, however, the usage (OS or CF) for the target. And a partition with usage type FW can only be copied to a FW partition.
2. Map those CHPIDs which have the source partition in their access or candidate list to CHPIDs on the target processor or channel subsystem.
3. Decide whether you want to move the physical connections from the source to the target by choosing either the **Upgrade** or the **Repeat** scenario.

   Note: For a better understanding of the **Repeat** and **Upgrade** scenarios and the wizard 's panel flow depending from your selection, refer to “General concepts of the copy wizards” on page 99.
4. If you selected the **Repeat** scenario, perform the following :
   a. You need to assign connections between the source CHPIDs and switch ports to new corresponding connections in the target.
   b. You need to assign connections of the source CHPIDs to new controller interfaces to the target.
5. Examine the summary of the partition information that is to be copied. Click **Finish** to perform the operation.

The wizard displays appropriate dialogs as you proceed through these steps. For further information about how to complete these dialogs, refer to the online help that is available for each wizard page. Also refer to “How HCM handles connections when using the copy wizards” on page 100.

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**.

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 29.

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.

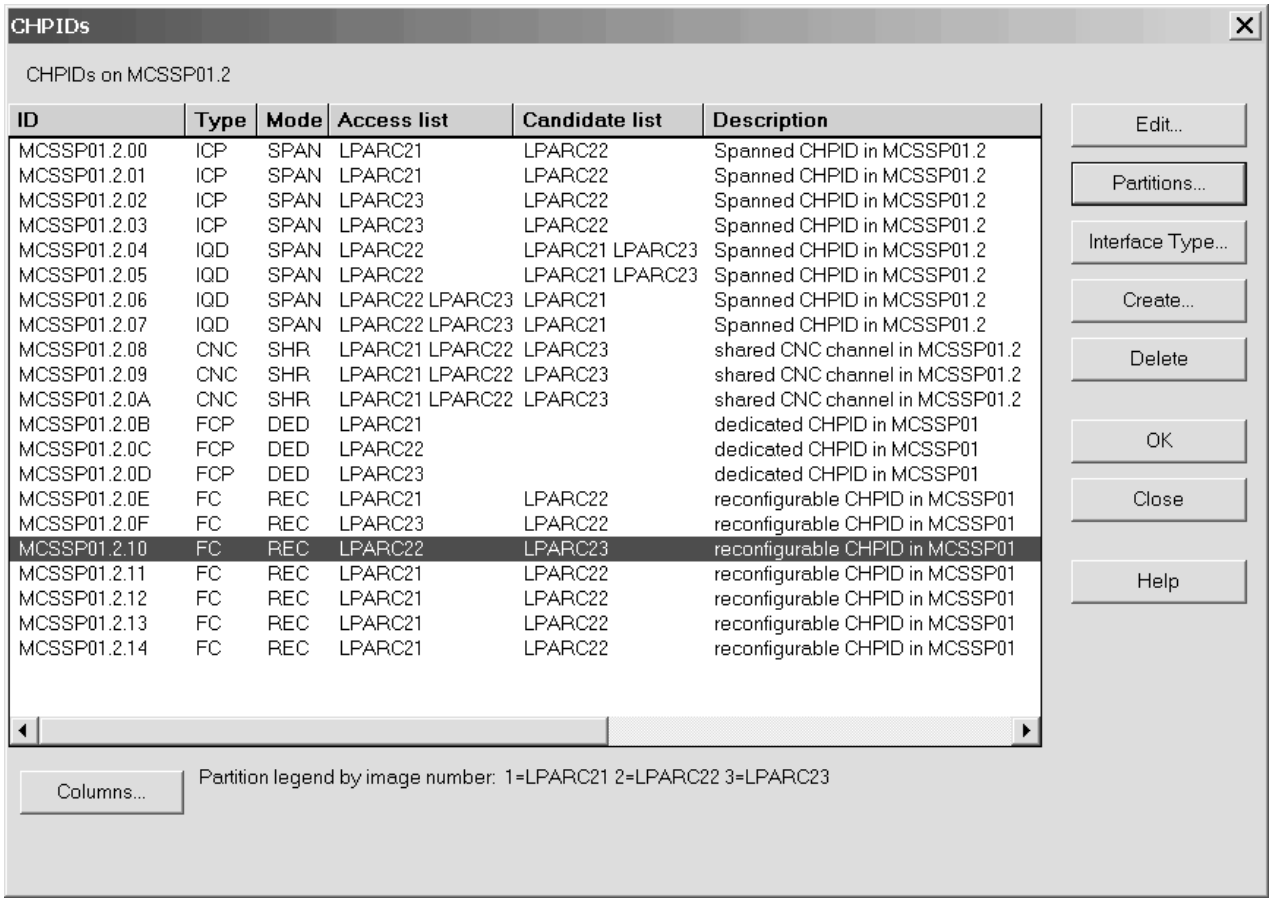


Figure 55. CHPIDs

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.

Pressing the **Columns...** button opens the **Choose Columns** dialog which lets you add or remove columns to or from the **CHPIDs** dialog. Figure 56 on page 119 shows how the columns have been selected that are displayed in Figure 55 on page 118. In the left hand box, you can read the columns that are additionally available, but currently excluded from the **CHPIDs** dialog.

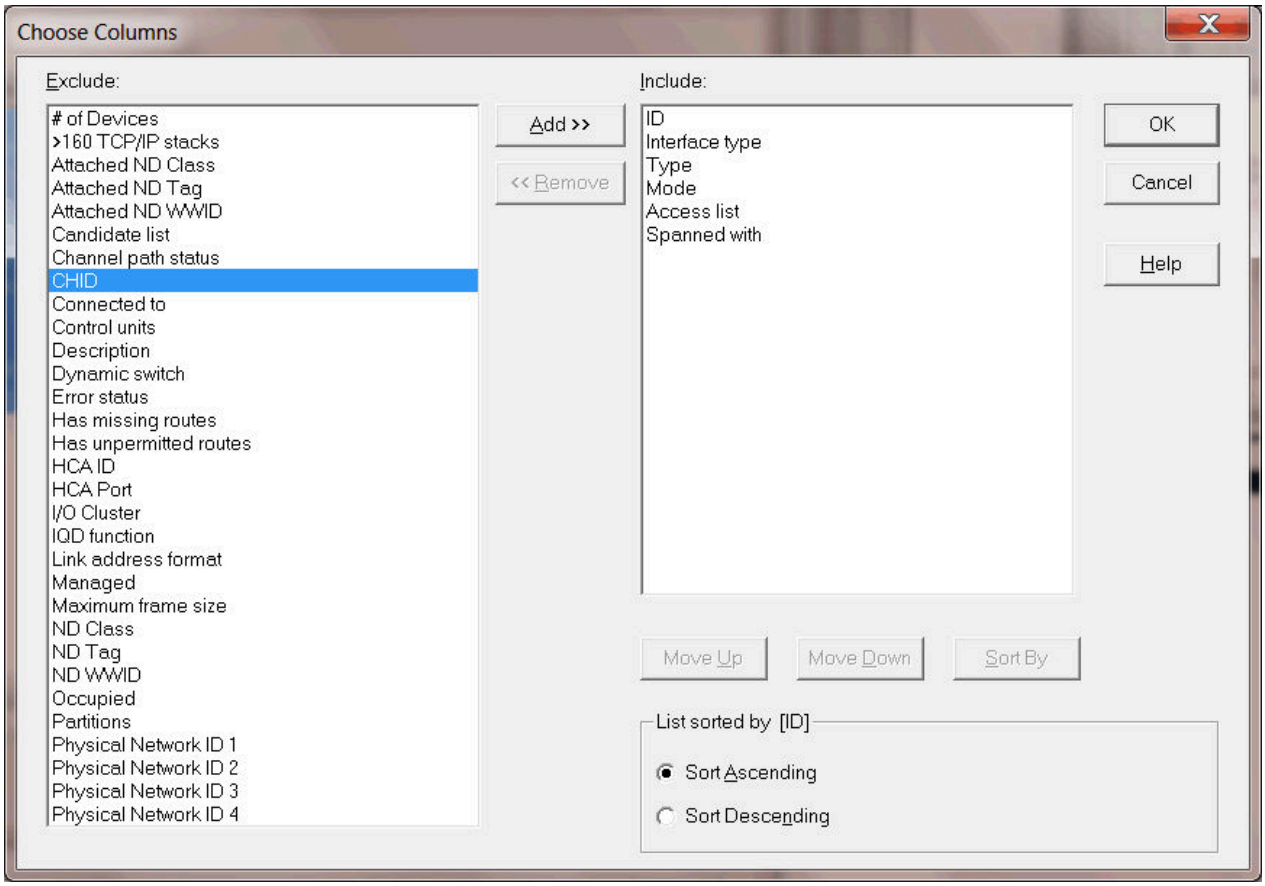


Figure 56. Choose Columns

There may be four special cases of CHPID text displayed in the list box, for example:

ID	Interface type	Type	Mode	Meaning
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, change the CHPID interface type:

- Go to the **CHPIDs** dialog
  - either via the **CHPIDs** button in the Channel Subsystems section of the **Processor** dialog or
  - by directly double-clicking a channel subsystem's CHPIDs from a **Processor** context menu.
- In the **CHPIDs** dialog, press the **Interface Type...** button to reach the **Change Path Interface Type** 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 are 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 and 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 online 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 can 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 is 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 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 or candidate list.

Candidate	Access	Operation mode
		Dedicated (DED)
		Reconfigurable (REC)
		Shared (SHR)
		Spanned (SPAN)

Figure 57. 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 opens, 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.

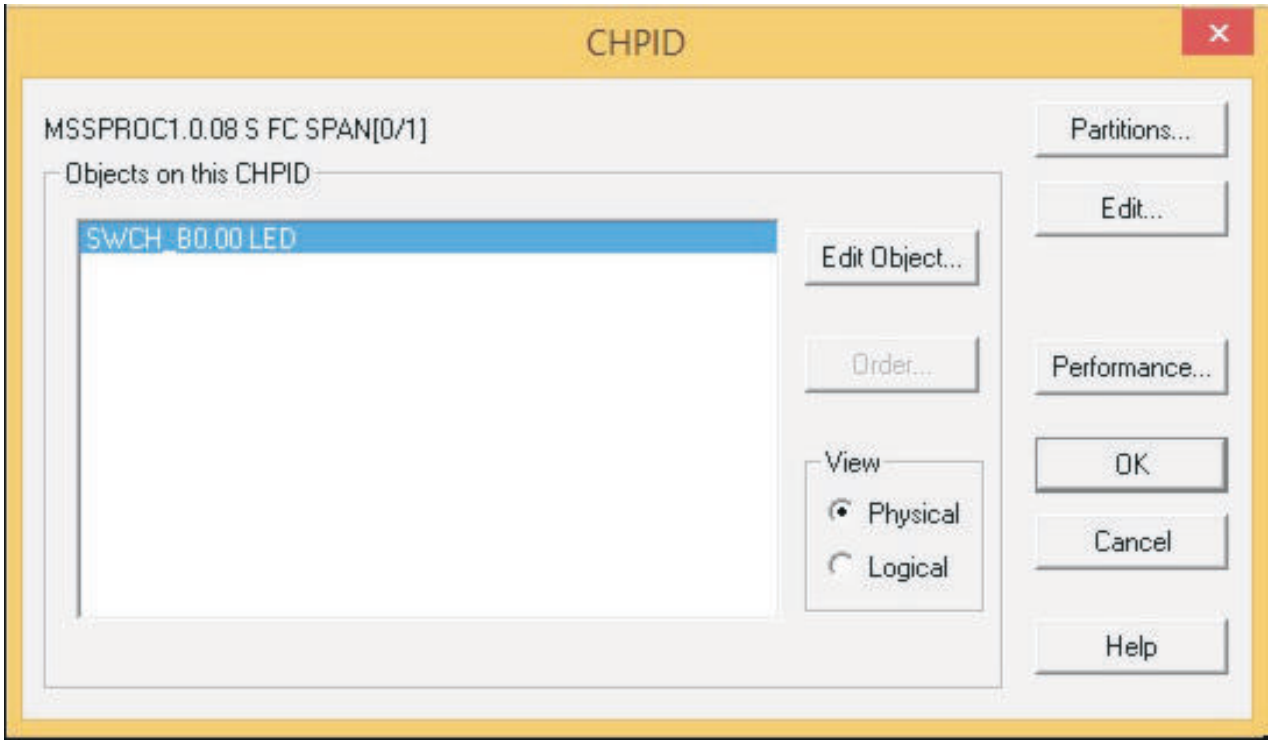


Figure 58. CHPID

Furthermore, you can retrieve the following details:

- The **Performance...** button lets you access performance data provided by RMF for the current CHPID.

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 opens. 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.

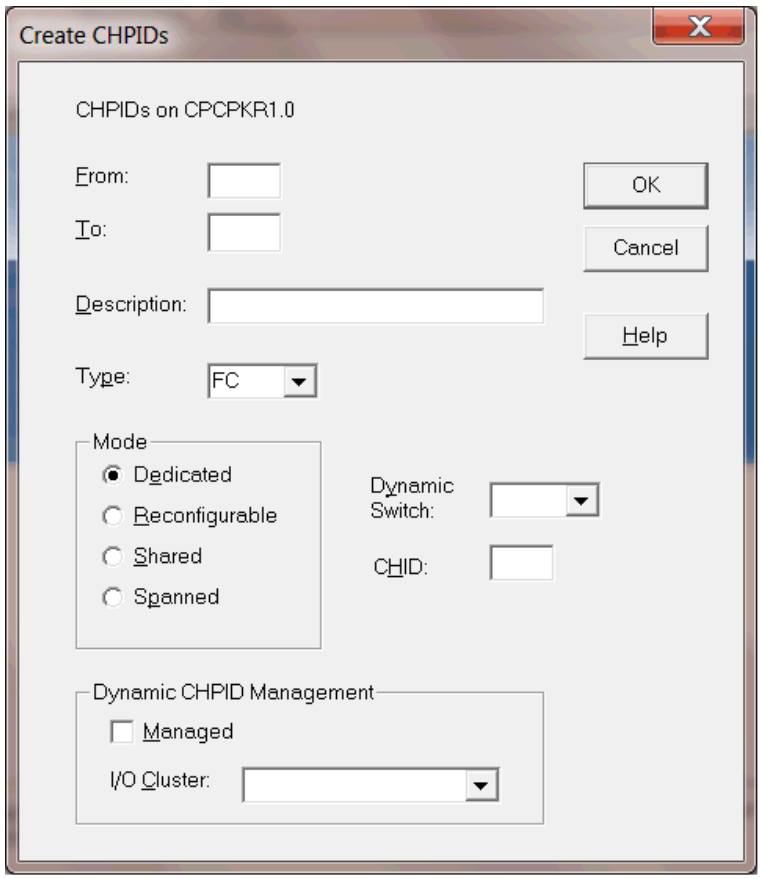


Figure 59. Create CHPIDs

You typically create a range of CHPIDs with the **From** and **To** fields. If you know this value, you can enter it into the CHID field. If you create a range of CHPIDs, the PCHIDs will be assigned sequentially, starting with the value you entered. For more information about the CHPID Mapping Tool refer to the *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.

Defining IQD channel parameters

For an IQD channel path type, HCM allows you to specify the following parameters:

maximum frame size
HCM allows you to specify a maximum frame size to be used for IQDIO requests on that channel path. For further information refer to *z/OS Communications Server: IP Configuration Guide*

IQD function
Starting with processor types 2817 and 2818, the Internal Queued Direct I/O (IQDIO) functions or the HiperSockets functions offers two new options:

- integration with the intraensemble data network (IEDN) controlled by the zManager functions which provide access controls, virtualization and management functions necessary to secure and manage the IEDN. This functionality is called extended IQD (IQDX).
- bridging an IQD channel to an external (customer managed) network

To support these functions, for IQD channels, HCD offers three choices:

Basic HiperSockets
The IQD channel path is connected to the internal HiperSockets network and is used without connection to the IEDN or an external network. This is the default.

IEDN Access (IQDX)
The IQD channel path supports IEDN via the Internal Queued Direct I/O Extensions (IQDX) function.

External Bridge
The IQD channel path works in basic HiperSockets mode and can be transparently bridged to an external (customer managed) network via the z/VM Virtual Switch bridge support.

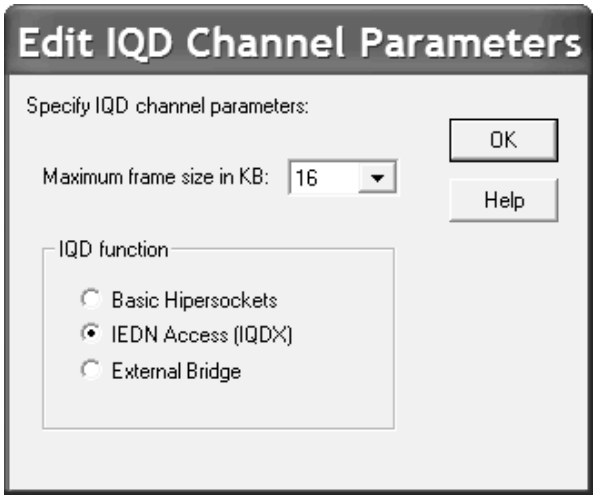


Figure 60. Edit IQD Channel Parameters

Defining more than 160 TCP/IP stacks

When defining or changing channel paths of type OSD, OSM, or OSX for processors with the corresponding support level, HCM prompts you with a dialog whether you want to define more than 160 TCP/IP stacks with this channel (Figure 61 on page 124). In this dialog, select the related checkbox if you want to have more than 160 TCP/IP stacks for the current channel. This choice, however, will disable priority queuing.

OSM channels require that more than 160 TCP/IP stacks are allowed.

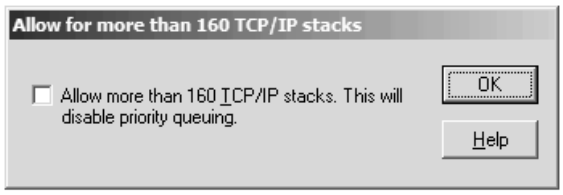


Figure 61. Allow for more than 160 TCP/IP stacks

Defining or editing a CIB channel

A channel path type CIB is defined for coupling over InfiniBand, where a peer coupling link is emulated on a host communication adapter (HCA). When defining or changing a CIB channel path, HCM prompts you with a dialog which asks for the specification of the ***Adapter ID*** of the HCA and the ***Port*** on the HCA of that channel path.



Figure 62. Specify HCA attributes

Defining or editing a CL5 channel

When defining or changing a CL5 channel path, HCM prompts you with a dialog that asks for the specification of the **Coupling Port** of that channel path.

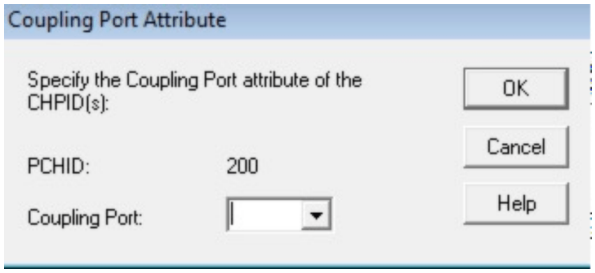


Figure 63. Specify Coupling Port attribute

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 with 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.

You can also over-define CIB channel paths for your OS partition. This avoids an outage for the definition of new CIB connections between an OS partition and a stand-alone coupling facility CPC.

To distinguish an over-defined CHPID from a physically installed CHPID, use character * for the PCHID value or the HCA ID for CIB channels, when over-defining the CHPID. An over-defined CHPID must adhere to all validation rules.

When installing the channel path later, you must edit the CHPID and replace the * by its valid PCHID or HCA ID.

Over-defined channel paths are not taken into account by an IOCDS download, by an IOCP build and by a dynamic activation of an I/O configuration. If a control unit contains only CHPIDs with a PCHID value or HCA ID *, the whole control unit (including any attached devices) is omitted from the configuration.

If a CHPID changes its PCHID or HCA ID from * to a valid value during a dynamic activation, an Add `CHPID` request is generated. Correspondingly, if the PCHID or HCA ID is changed from a valid value to an *, a `Delete CHPID` request is generated.

When building a CONFIGxx member, CHPIDs with a PCHID/HCA ID * are skipped. Attached control units including attached devices are also omitted.

When copying a configuration or generating I/O configuration statements, channel path definitions with PCHID=* and AID=* are included.

When building a production IODF, HCD requires that the CIB channels are connected, even if they are over-defined. If the connection is within the same processor, a mix of over-defined and fully-defined HCA IDs is not accepted. In this case, error message CBDG541I is issued and the production IODF is not built.

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 **Edit a CHPID** dialog. You can include and exclude columns 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 119 for more details). Note that you can change the CHPID type while the CHPID is connected.

When you change the CHPID type from 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 appears 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 193 for more details.

Note:

1. For OSD CHPIDs, the **Edit CHPID** dialog contains a checkbox which allows you to define more than 160 TCP/IP stacks for this channel.
2. For IQD CHPIDs, the **Edit CHPID** dialog contains
   - a drop down box that lets you define the maximum frame size in kilobytes
   - a group of radio buttons to select the IQD function.

   For more information, see “Defining more than 160 TCP/IP stacks” on page 124 and “Defining IQD channel parameters” on page 123.

Deleting a CHPID

Select one or more CHPIDs from the list box in the **CHPIDs** dialog and click on the **Delete** button. HCM displays 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 is 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 64 on page 127) 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 is filled with the partitions of the corresponding channel subsystem. 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 and eligible channel subsystems. Channel subsystems are eligible if they have no different channel path defined with the same channel path identifier as the one you are currently working with.

Examples:

Let us assume you have a processor with three channel subsystems named CSS0, CSS1, and CSS2. Let us further assume that the following shared CHPIDs are already defined: CHPID 10 and 20 to CSS0, CHPID 10 to CSS1, and no CHPID yet to CSS2. If you want to define a new spanned CHPID 30, you can span this new CHPID over all three channel subsystems, as there is no CHPID 30 yet defined in any channel subsystem. Hence, the **Assign Partitions** dialog contains all partitions defined in CSS0, CSS1, and CSS2. If you want to define a new spanned CHPID 20 in CSS2, you can only span it over CSS2 and CSS1, as there is already a CHPID 20 defined in CSS0. Therefore, the **Assign Partitions** dialog is filled with partitions from CSS 1 and CSS2, while no partition from CSS0 is shown in the dialog. Finally, if you want to define a new CHPID 10 in CSS2, you can only define it as SHR (or DED / REC), but not as spanned, as there are already CHPIDs 10 defined to both CSS0 and CSS1. In this case, the **Assign Partitions** dialog only contains partitions from CSS2, but not from the other two 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 can 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.

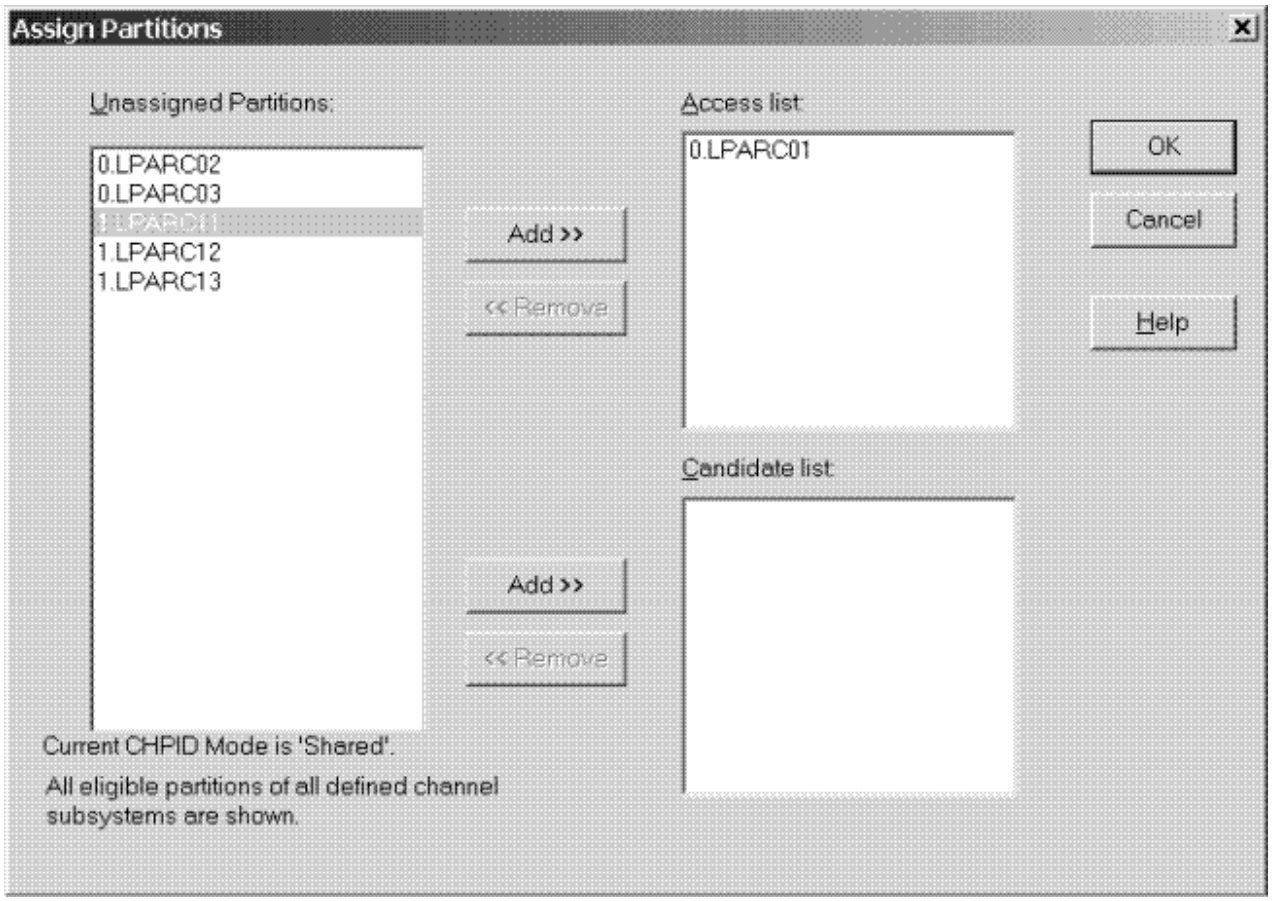


Figure 64. Assign Partitions

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 64 on page 127, 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 65 on page 129, 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:

```
MCSSPRO1.0.18 [31D] S FC SPAN[0/1] to SWCH B0.06
```

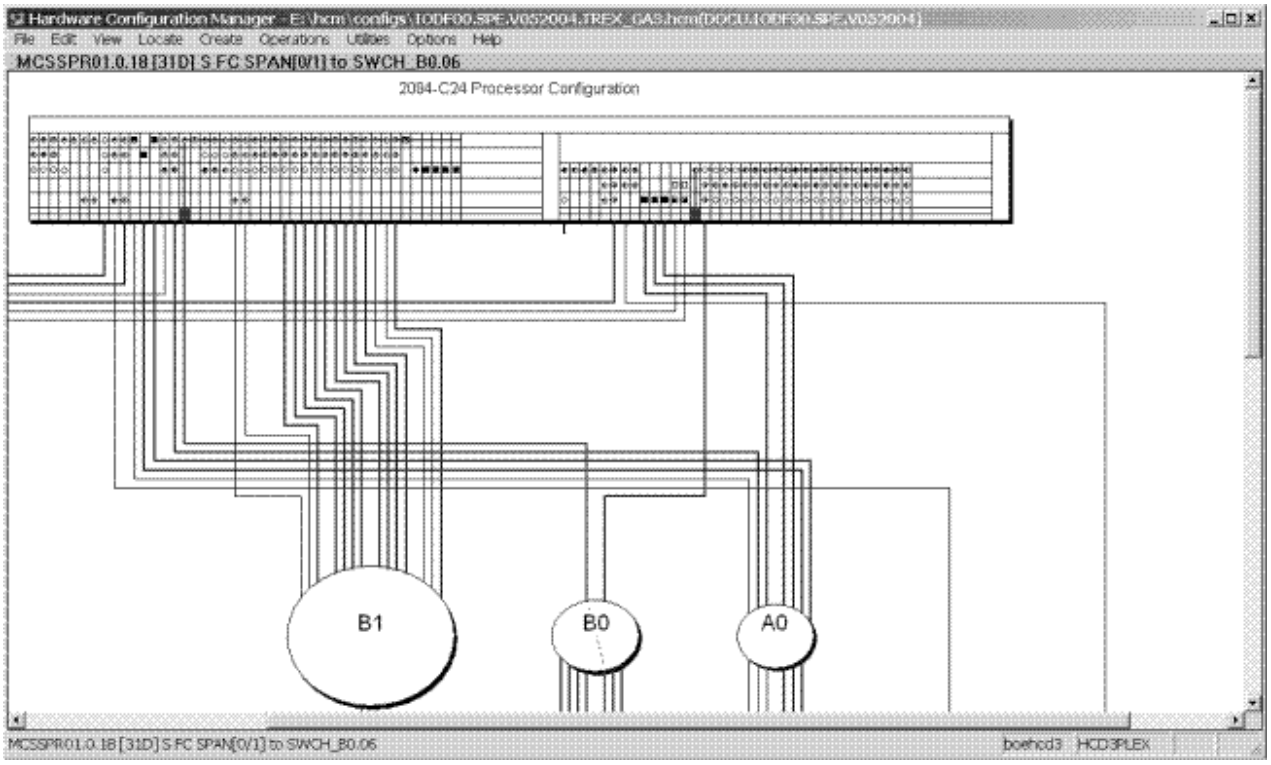


Figure 65. Spanned CHPIDs in the Configuration Diagram

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 66 on page 129:

- 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.

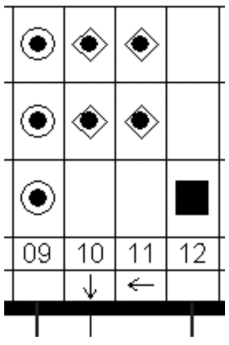


Figure 66. Physical connection information for spanned CHPIDs

Navigating with spanned CHPIDs

The context 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 185
- "Assigning CHPIDs to partitions" on page 126

Please also read the information contained in "Physical and logical aspects of spanned CHPIDs" on page 128.

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 233 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 157 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 183.

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 132). You can select a string to edit, or create a string under the selected controller. See "Strings" on page 140 for more information.

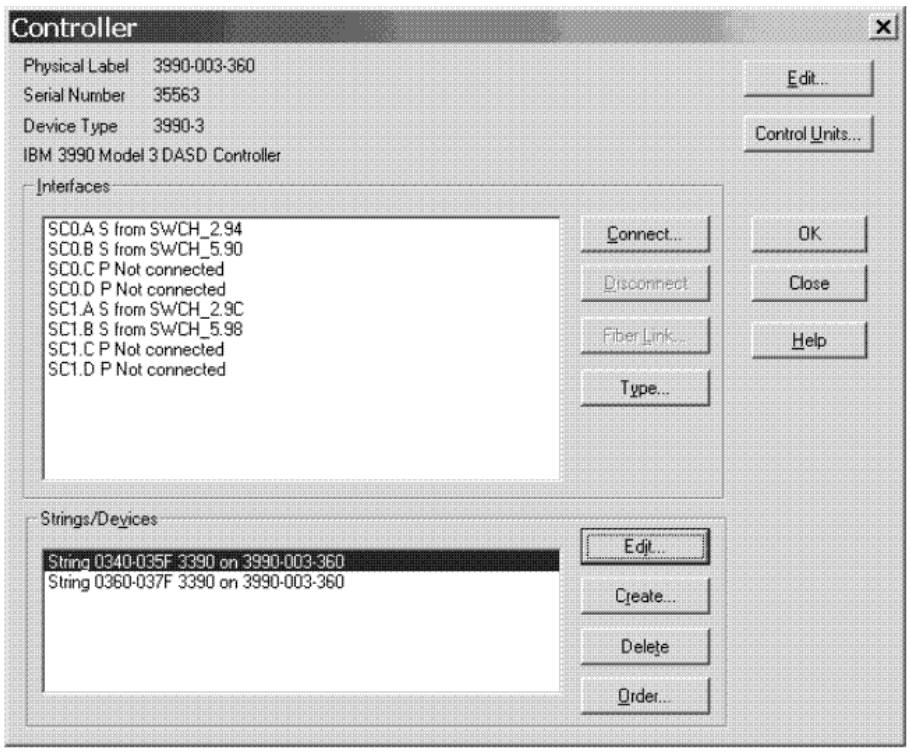


Figure 67. Controller

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.

Note: HCM offers a wizard that lets you create a complete I/O subsystem consisting of

- controllers (with segments and control units) and
- strings (with devices).

See "Creating/Copying/Editing an I/O subsystem" on page 234 for more information.

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 15, "Physical description files (PDF)," on page 379 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" on page 132). Select a physical description and press **OK** to create the new controller.

Changing the physical description affects all controllers within a subsystem (for example, 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** displays 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 is 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, HCM opens a dialog 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 is 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 33 for details).

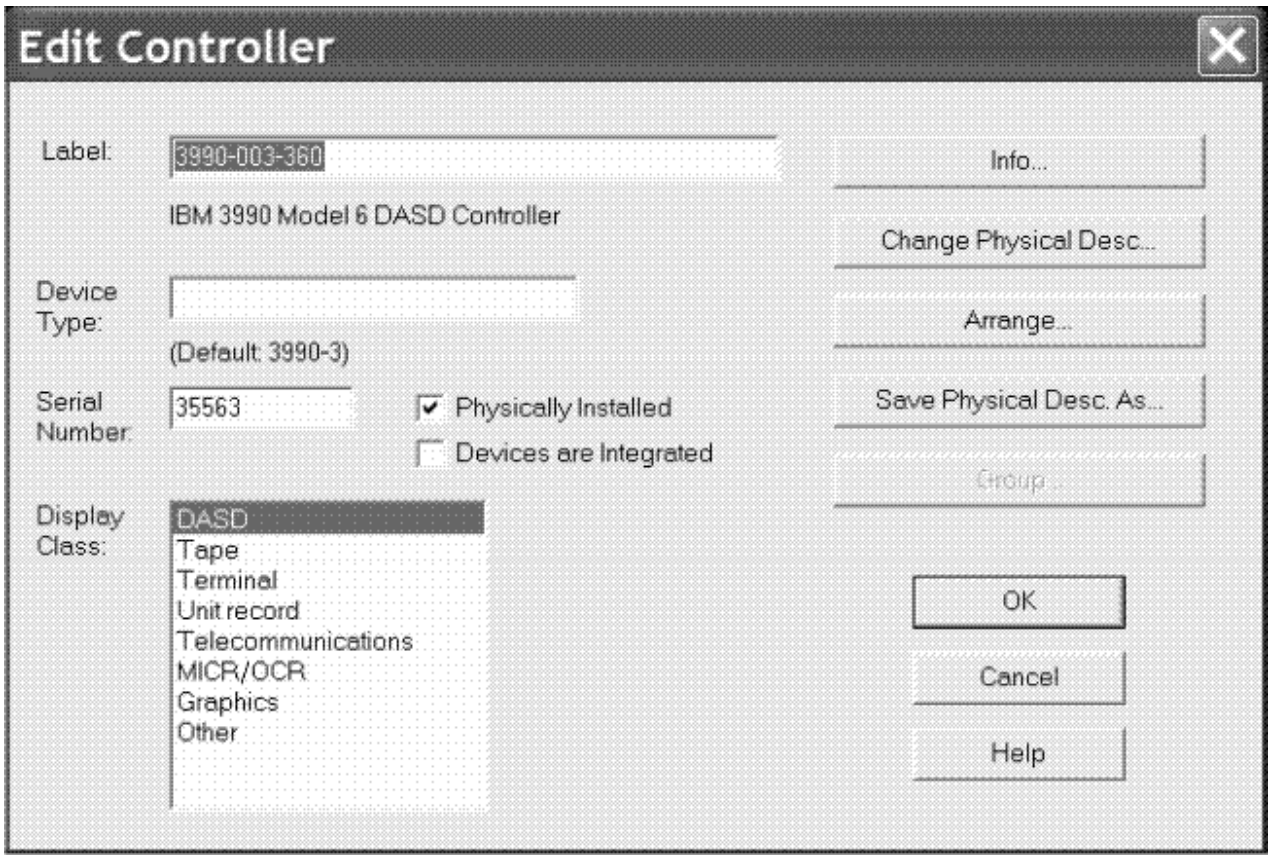


Figure 68. Edit Controller

Creating a new physical description file for a controller

You can save the current layout and properties of the edited controller as a model for further controllers by creating a new physical description file (PDF).

To create a new PDF for a controller, use the **Save Physical Desc. As...** button. This button opens the **Specify Subsystem Name** dialog, into which you must enter a unique subsystem name used by HCM to distinguish PDFs . Pressing **OK** in this dialog lets you save the PDF with a specified file name.

Note: The new PDFs are only available after loading a new or reloading the current HCM configuration file or after an HCM restart.

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 29.

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 131.

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 137).

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.

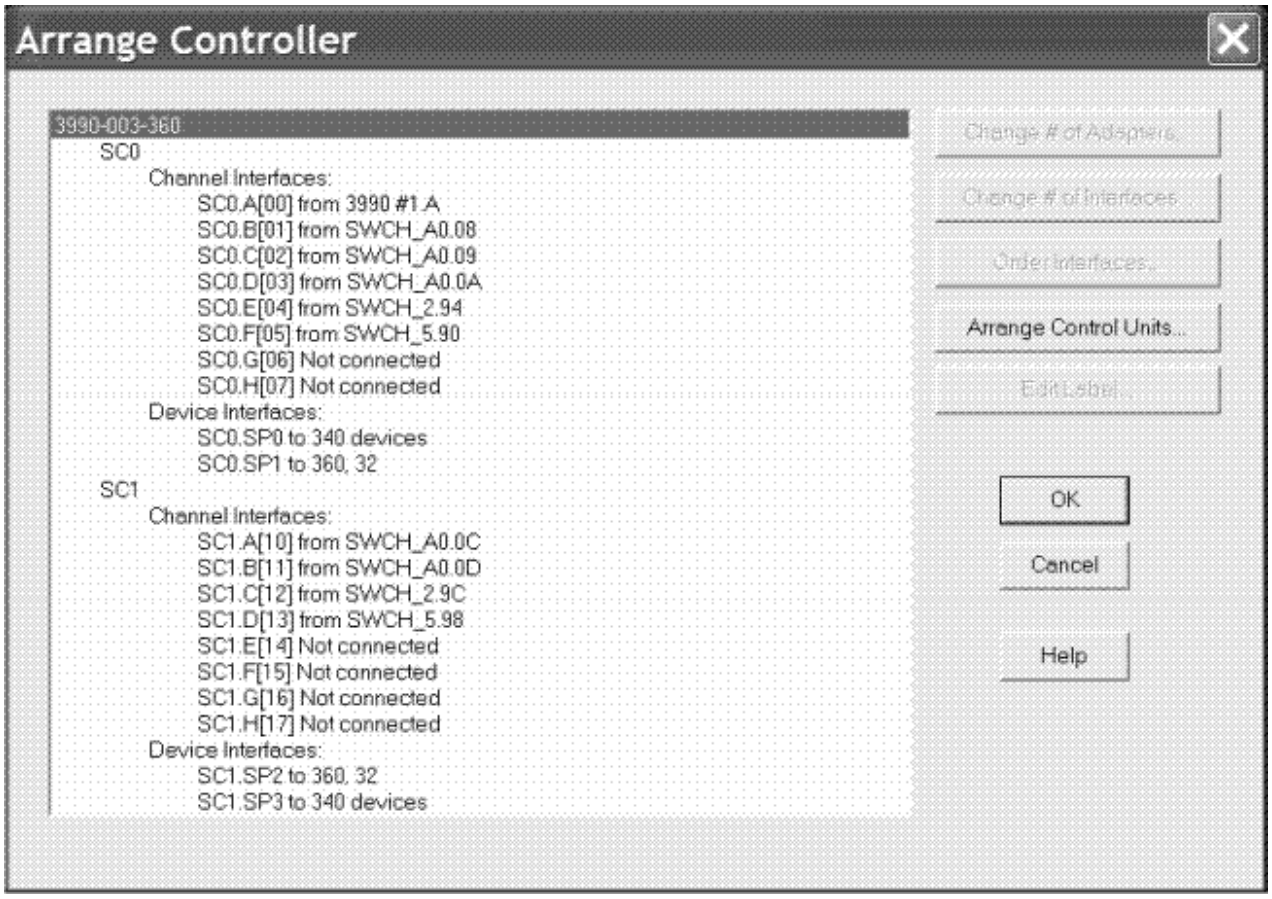


Figure 69. Arrange Controller

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 69 on page 134 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 adapters

Select a segment (for example, SC0) from the list in the **Arrange Controller** dialog and press the **Change # of Adapters...** button. The **Number of Adapters** dialog appears.

Use this dialog to change the number of channel or device adapters in the selected controller segment. New adapters are created with the default number of interfaces and are added to the end of the segment. Also, HCM assigns the default label “DA” to an adapter, if the adapter has no label specified in the physical description file (PDF).

An adapter cannot be deleted if any of its interfaces are connected. 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.

If you reduce the number of adapters, HCM removes them from the end of the segment. If you want to remove unused adapters from the front of the segment, you must reorder the adapters so that all unused adapters are grouped together at the end of the segment. Then you can reduce the number of adapters.

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.

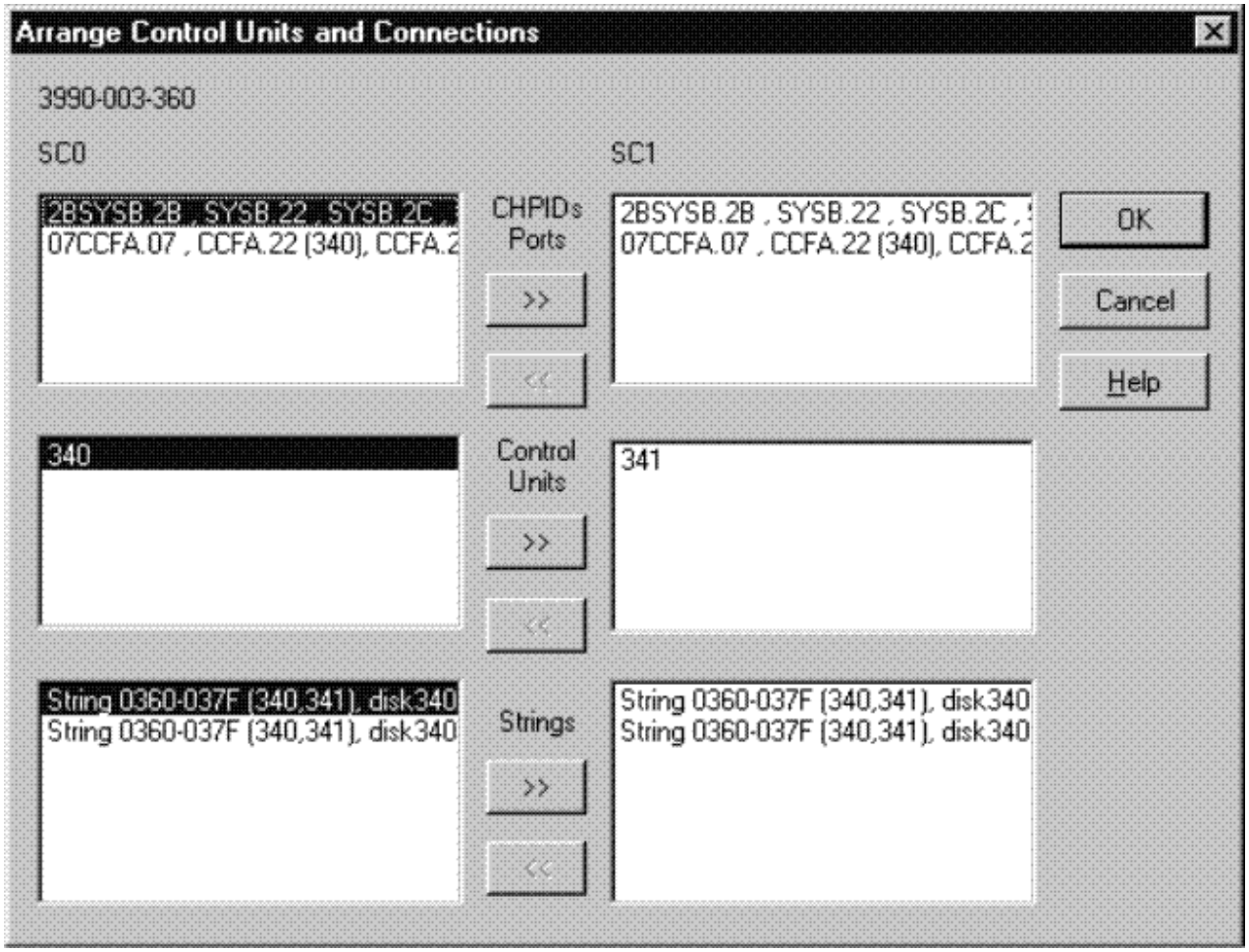


Figure 70. Arrange Control Units and Connections

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 135) to further correct the order of interfaces within each segment.

Editing labels of adapters and interfaces

With HCM, users may utilize more controller adapters, more controller adapter interfaces, more controller device adapters and more controller device adapter interfaces than being defined in the controller ' >s> physical description file. All adapters and interfaces without an explicit label provided by the corresponding physical description file, get a default name, for example, ‘CA’ for a controller adapter.
If you prefer your own naming for labels, HCM lets you modify adapter and interface labels. For this purpose, use the **Edit Label...** button. Depending on the selected object, one of the following dialogs is invoked:

Selected object	Invoked dialog
controller adapter	Edit Controller Adapter Label
controller adapter interface	Edit Controller Adapter Interface Label
controller device adapter	Edit Controller Device Adapter Label
controller device adapter interface	Edit Controller Device Adapter Interface Label

Each of these dialogs also contains a radio button that allows you to have unlabeled adapters or interfaces instead of default or user-specified labels.

Grouping controllers into subsystems

Some controller types can be grouped together in a subsystem, for example, 3990 dual frame controllers or 3803s linked via communication cables.

Click on the **Group...** button in the **Edit Controller** dialog to open the **Group Controllers** dialog (see Figure 71 on page 137). 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 makes 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.

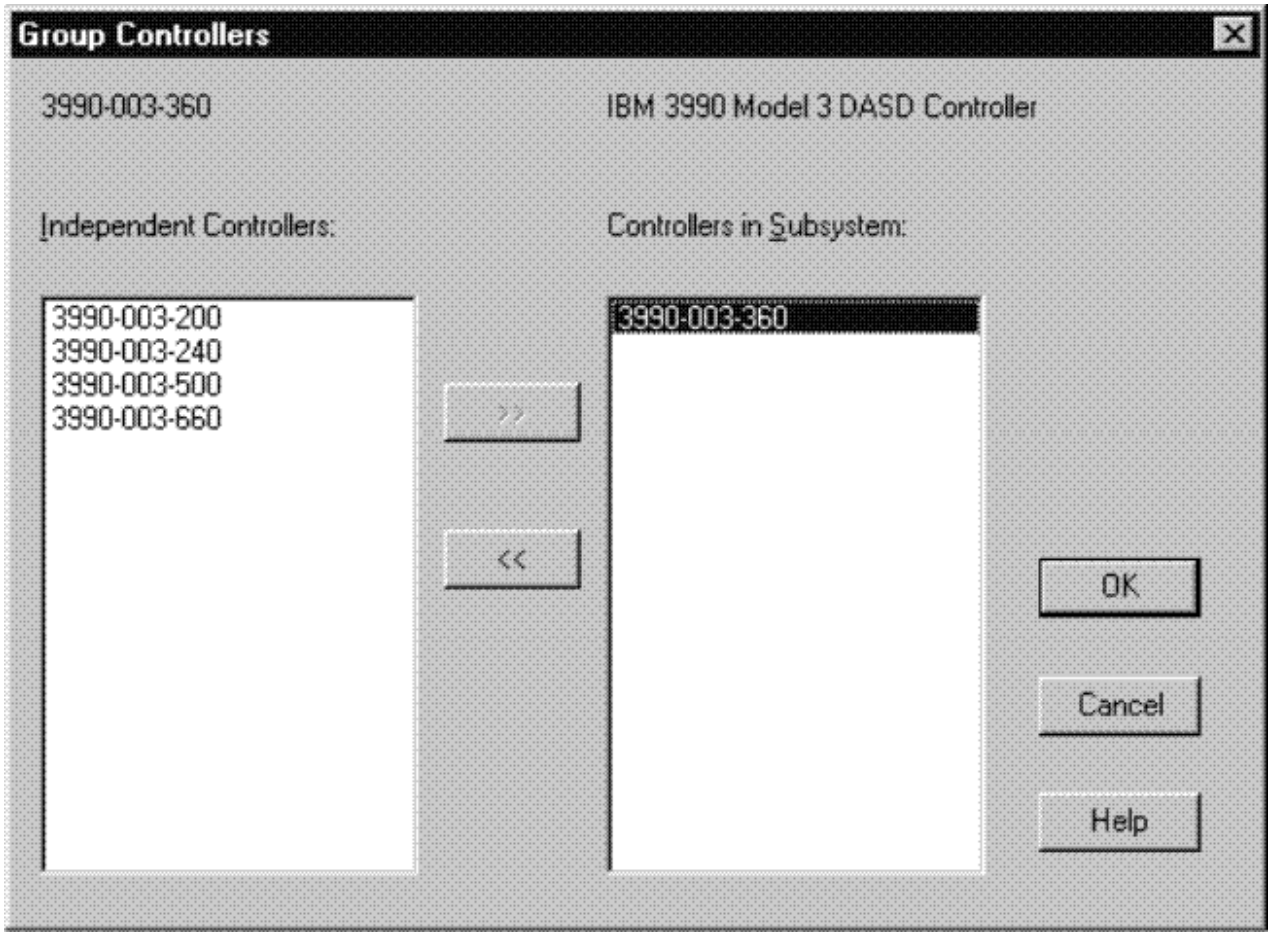


Figure 71. Group Controllers

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 215 for information about 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 67 on page 131). 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 117 for more details.

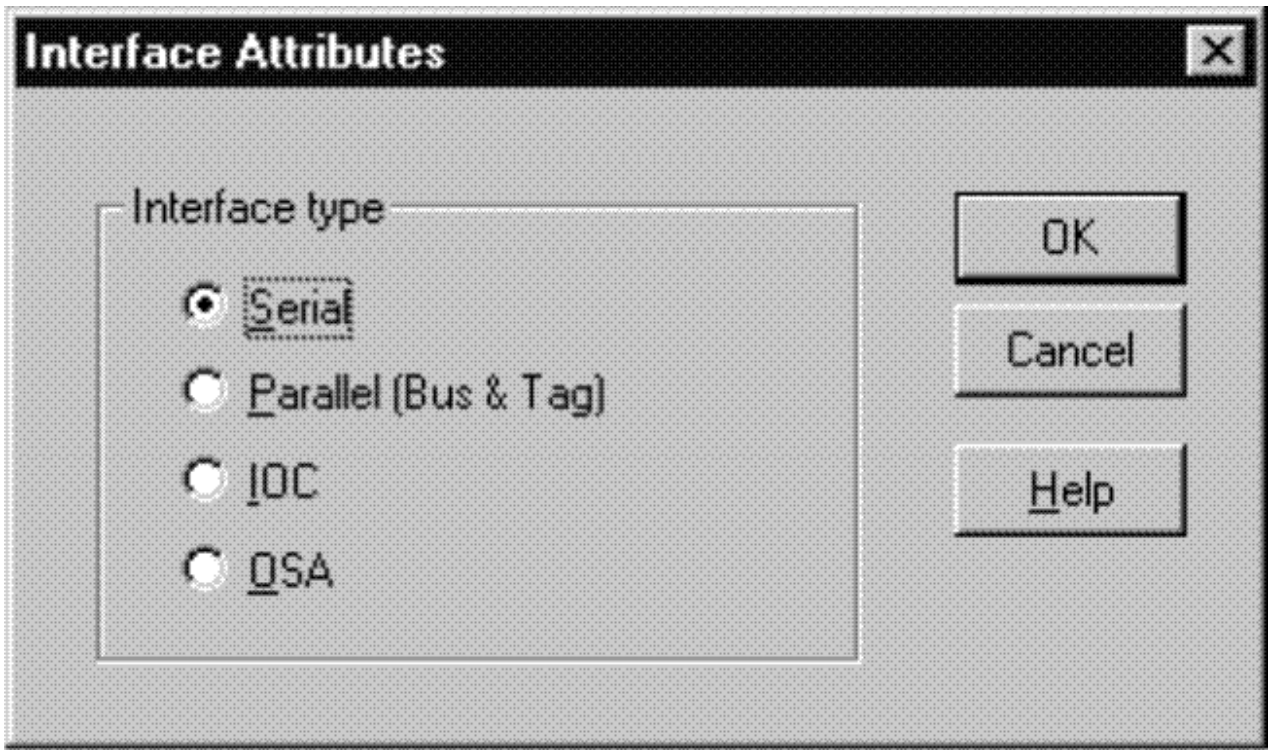


Figure 72. Interface Attribute

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 73 on page 139):

- 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 185 for details.

You can modify the switch control unit by accessing the **Controller** dialog. For more details, see “Editing switch port attributes” on page 157.

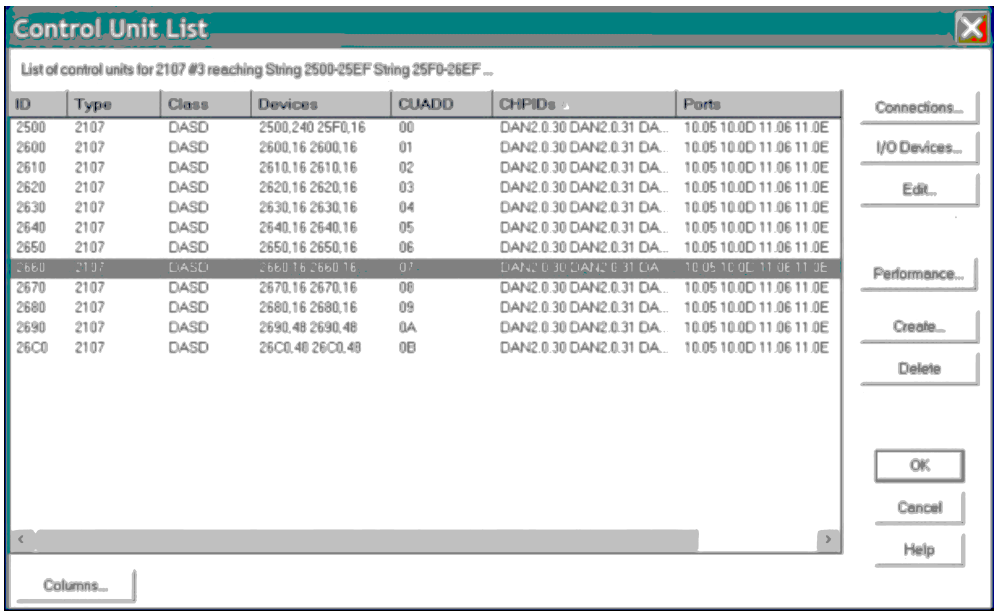


Figure 73. Control Unit List

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 134).

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.

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.

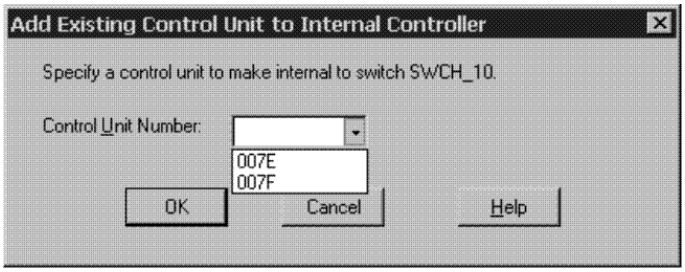


Figure 74. Add Existing Control Unit to Internal Controller

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 are also moved into the switch's internal string.

The controller which contained the switch-internal control unit will not be deleted after this control unit has been moved into the switch. If you do no longer need the controller, you can delete it manually.

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 177.

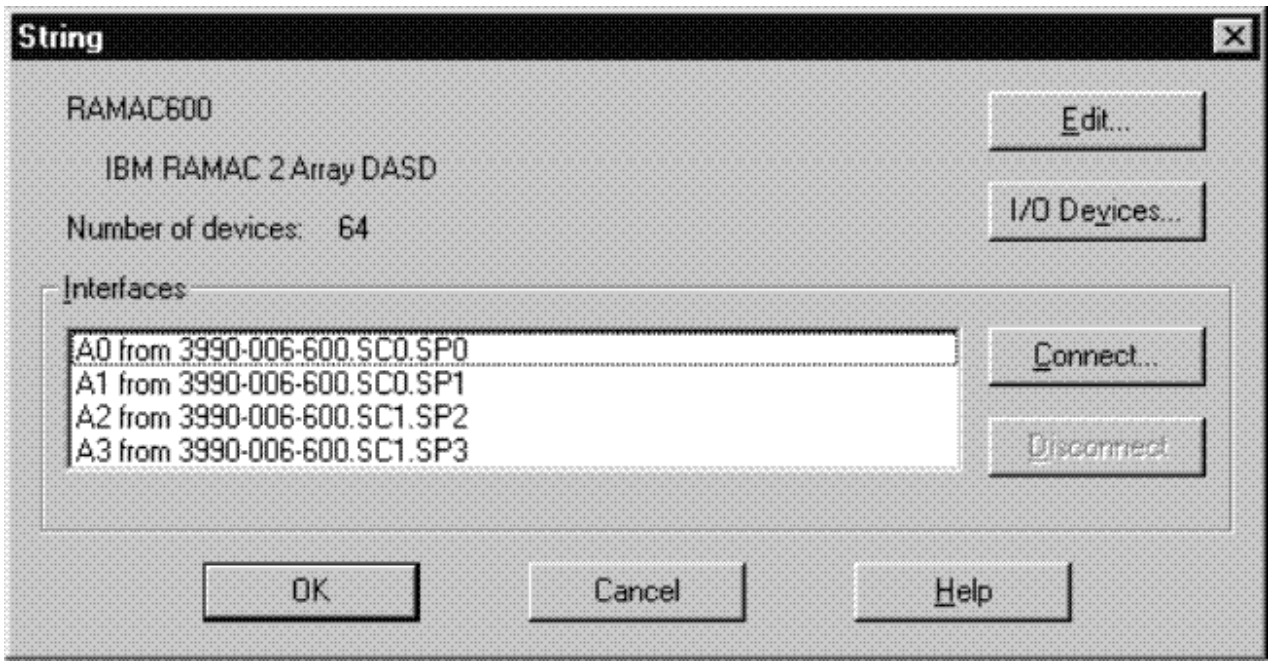


Figure 75. String

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.

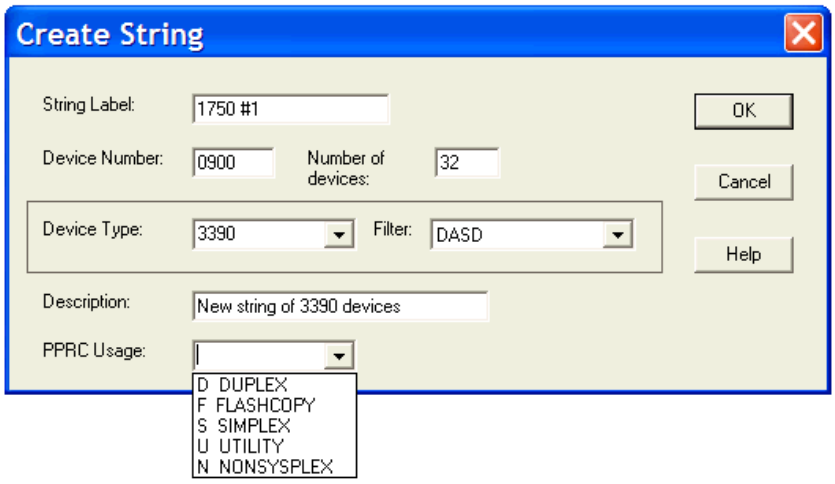


Figure 76. Create String

For DASD devices, you can optionally define a PPRC (peer-to-peer remote copy) usage type. A classification as FLASHCOPY, SIMPLEX, UTILITY, or NONSYSPLEX device is of descriptive character only. DUPLEX devices, attached to a primary operating system configuration with OFFLINE=YES are defined to an optional D/R site operating system configuration with OFFLINE=NO and vice versa, when the D/R site OS configuration is generated. For more information about PPRC usage type DUPLEX refer to “D/R site OS configurations” on page 246.

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 55 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** opens the **Select Controller Device Interface** dialog to allow you to connect the string to the controller.

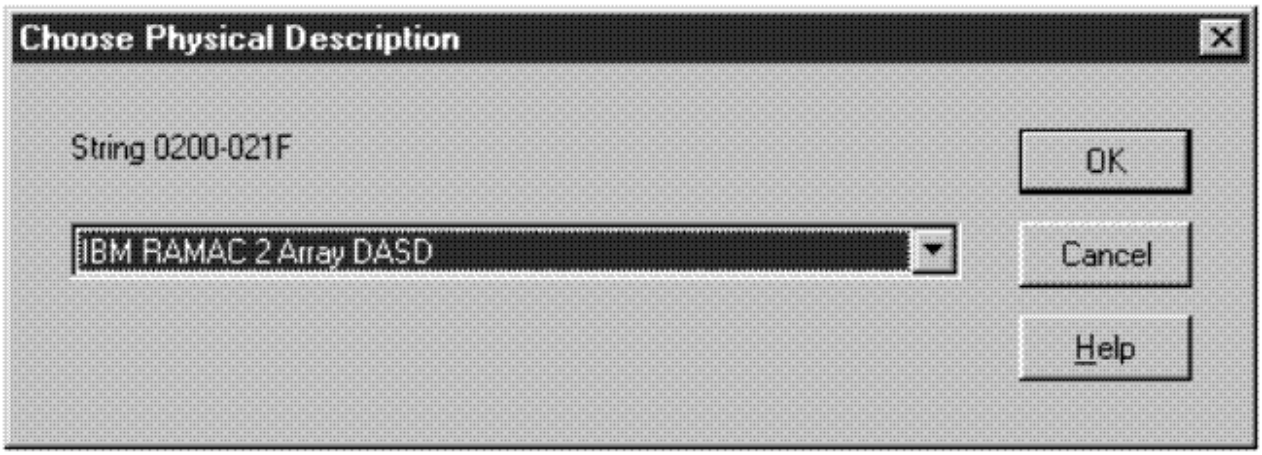


Figure 77. Physical Descriptions dialog

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 145 for more details.

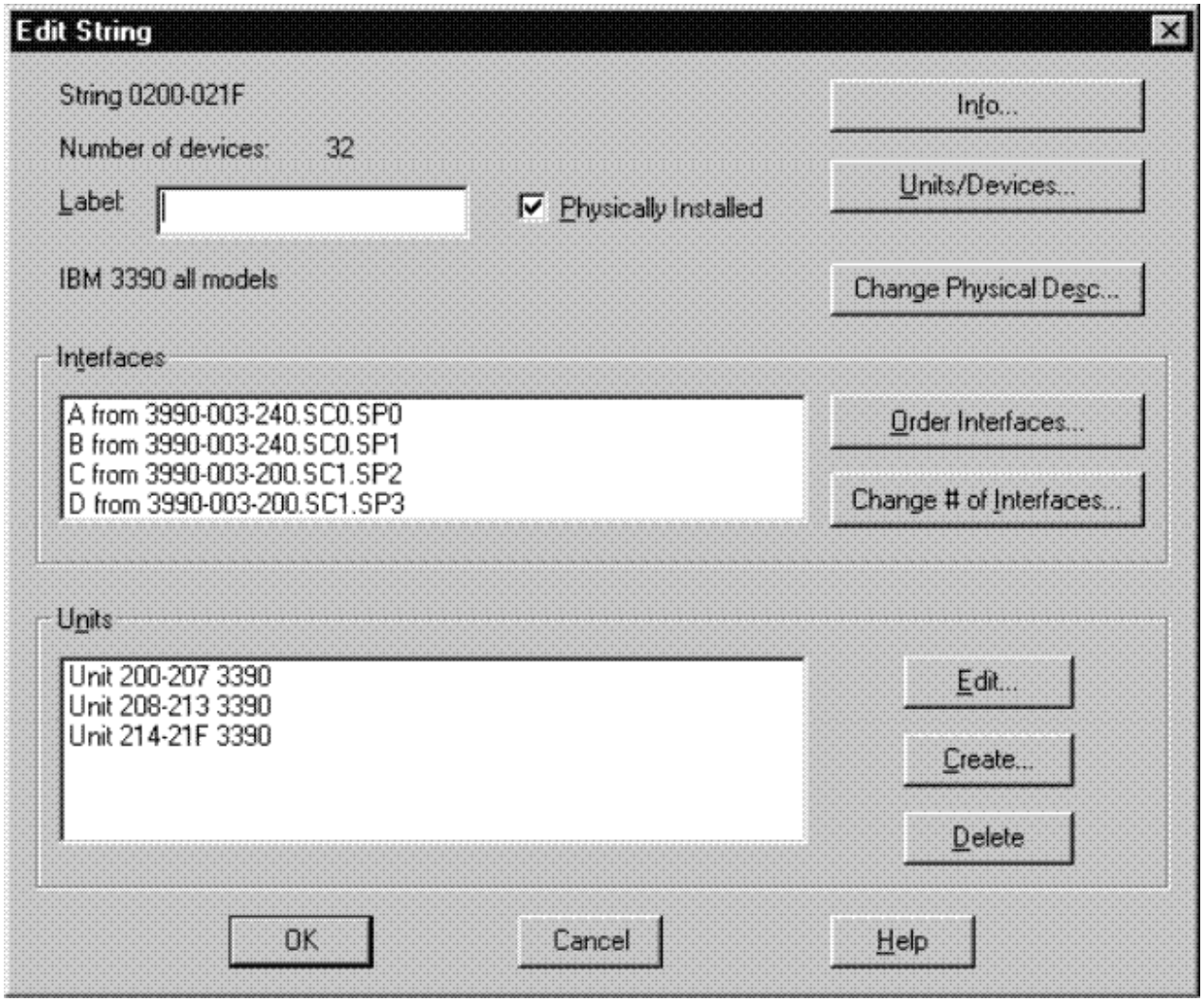


Figure 78. Edit String

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 29.

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.

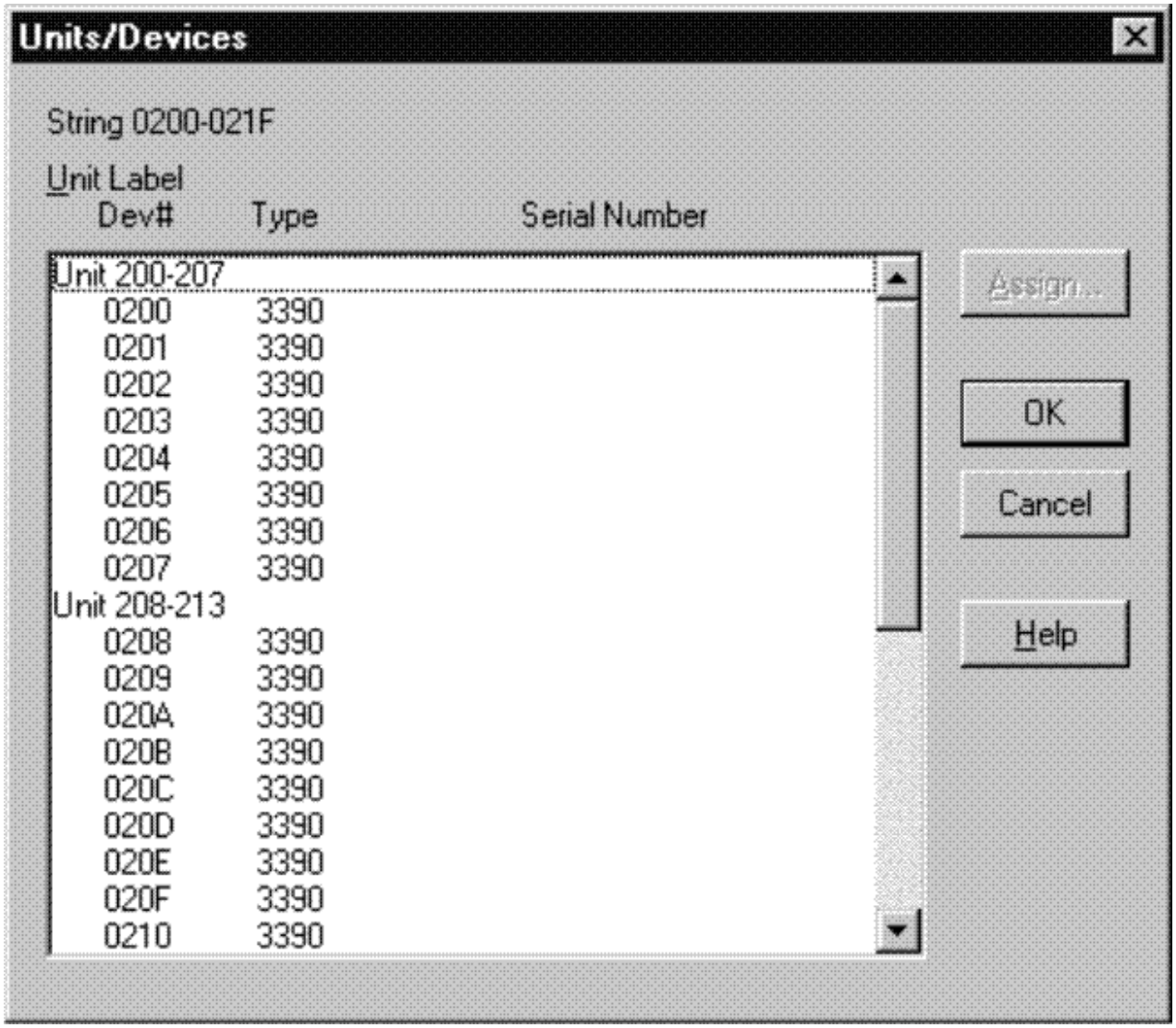


Figure 79. Units/Devices

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 55 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 46 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 143.

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.

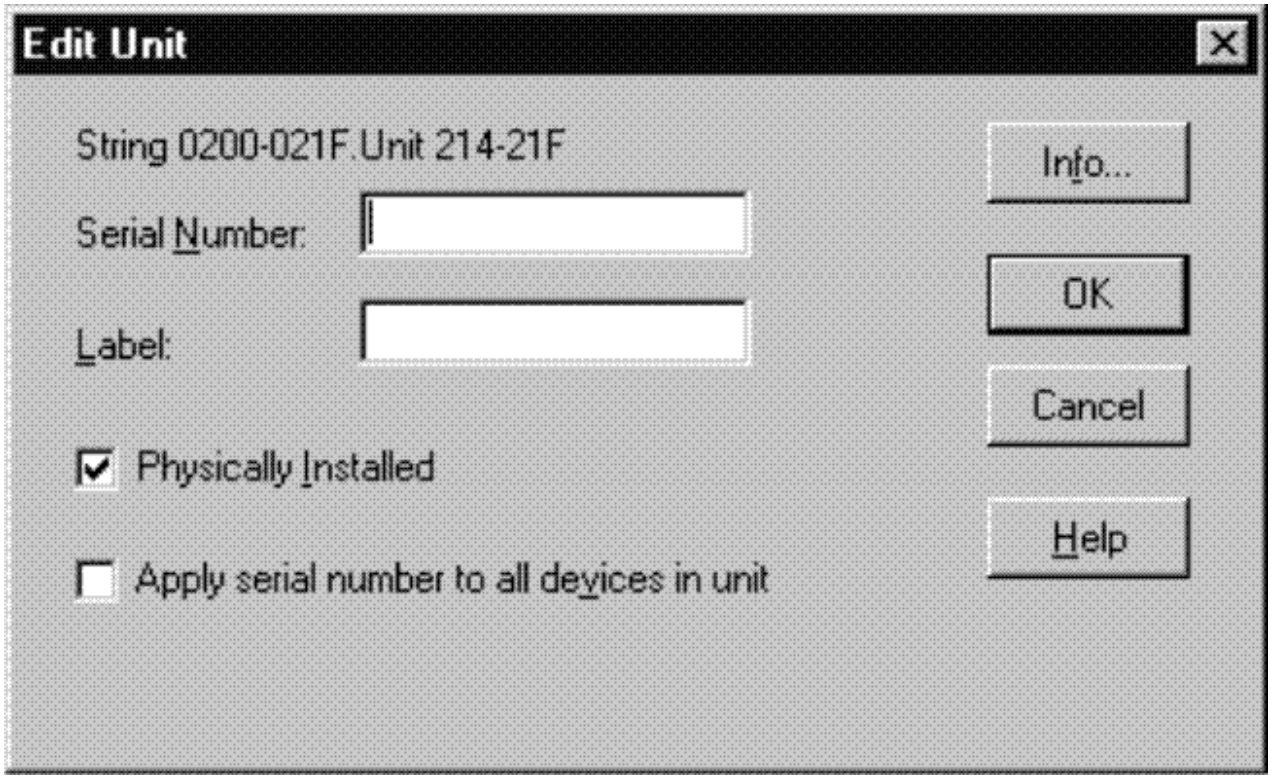


Figure 80. Edit Unit

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 29.

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 177 for details.

There are different possibilities to open the **I/O Device List**:

- Select a string from the configuration diagram and select **I/O Devices ...** from its context menu , or choose **String...** from the **Edit** menu, select a string, press **OK** and 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.

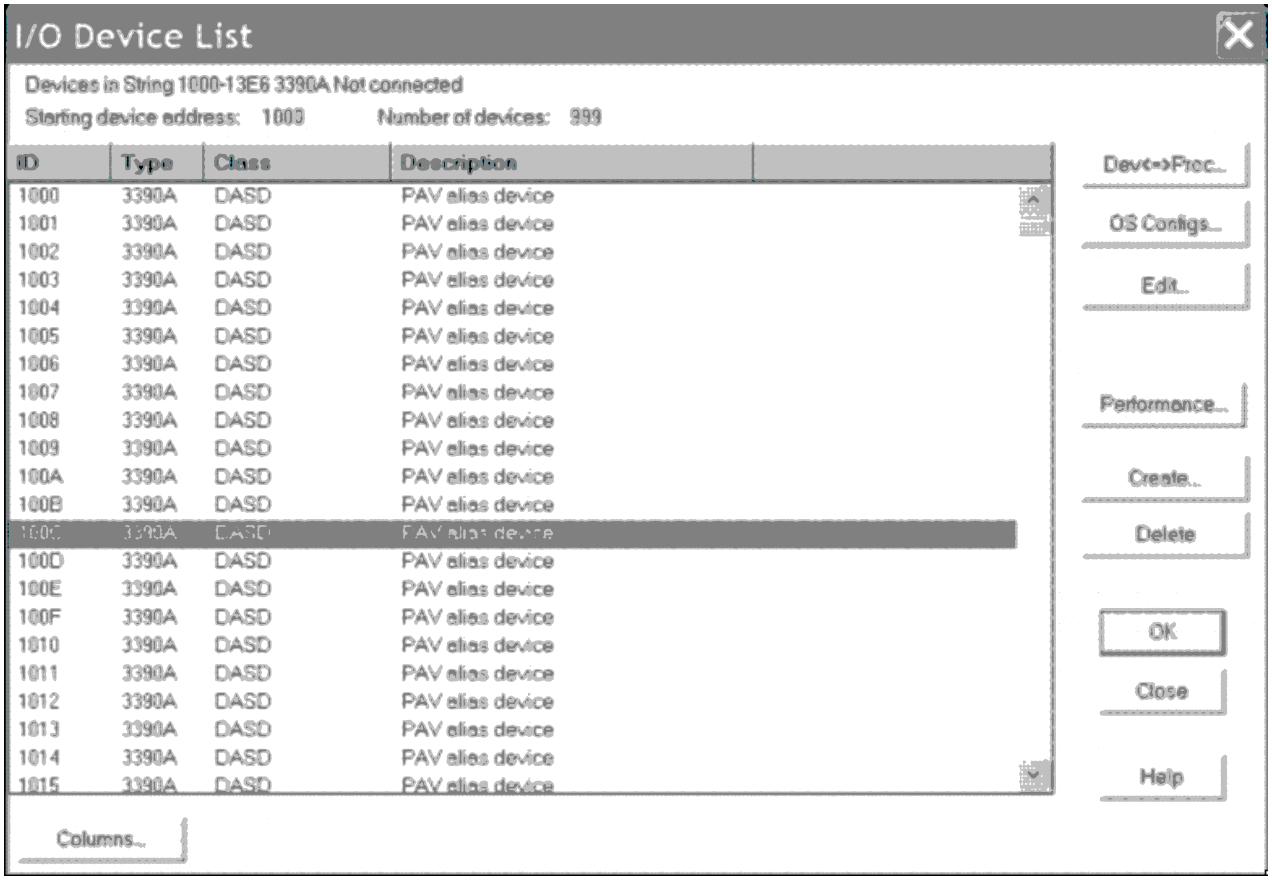


Figure 81. I/O Device List

The **I/O Device List** dialog allows you to display the serial numbers and VOLSERs, device descriptions, control unit numbers and further information about devices in the list box by clicking the ***Columns...*** button to include selected columns into the list.

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.

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 180.

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 156.

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 143 for more information.

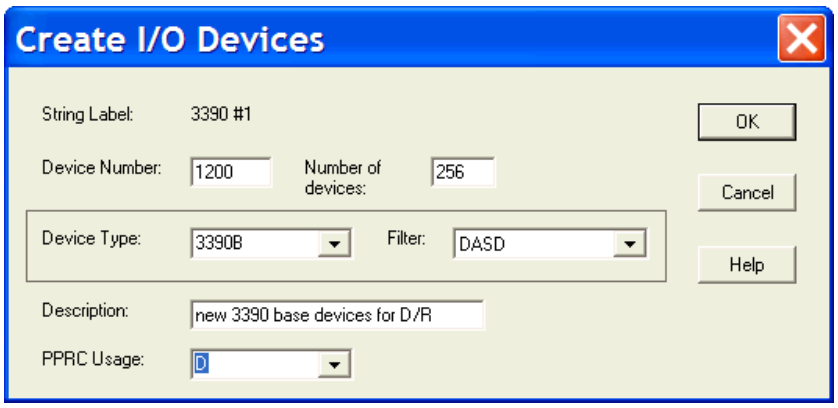


Figure 82. Create I/O Devices

In the ***PPRC Usage*** field, you can optionally define a PPRC usage type (peer-to-peer remote copy) for DASD devices. A classification as FLASHCOPY, SIMPLEX, UTILITY, or NONSYSPLEX device is of descriptive character only. Devices of PPRC usage type DUPLEX, attached to a primary operating system configuration with OFFLINE=YES are defined to a possible D/R site operating system configuration with OFFLINE=NO and vice versa. For a more information, refer to "D/R site OS configurations" on page 246.

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 177).

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 97.

Edit the information and click on **OK**.

Note:

1. 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.
2. If you want to change the device type of a device that is defined in a subchannel set other than 0, then the device type change will only be accepted if the target device type can also be defined in an alternative subchannel set.

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 250.

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) .

Changing the subchannel set for devices

Some scenarios may require to change the subchannel set placement of devices. For example, when you copy a channel subsystem from one processor to a target processor, but in the target processor you want to have a different placement of devices. Or, you may want to use your devices in a different placement in different operating systems. Therefore, you can change the placement of devices in regard to either one or both of the following aspects:

- **Device <-> Processor** relation
- **Device <-> Operating System** relation.

You have several possibilities to invoke the dialogs for changing the subchannel set ID for devices:

Modifying the subchannel set ID for 'Device <-> Processor' and 'Device <-> Operating System' at the same time:

Start from one of the following dialogs:

- From the **Edit** menu bar choice, select **Device...** to reach the **Edit a Device** dialog.
- From the **Locate** menu bar choice, select **Device...** to reach the **Locate One or More Devices** dialog.

Both dialogs contain the **Change SCHS** button which lets you change the subchannel set placement of (multiple) eligible devices. HCM first prompts you for which processor(s) and then for which operating system(s) you want to change the subchannel set:

1. In the **Object List for Subchannel Set ID Change** dialog for processors, you select the processor(s) for which you want to change the subchannel set ID for the device <-> processor definition and press **OK**.

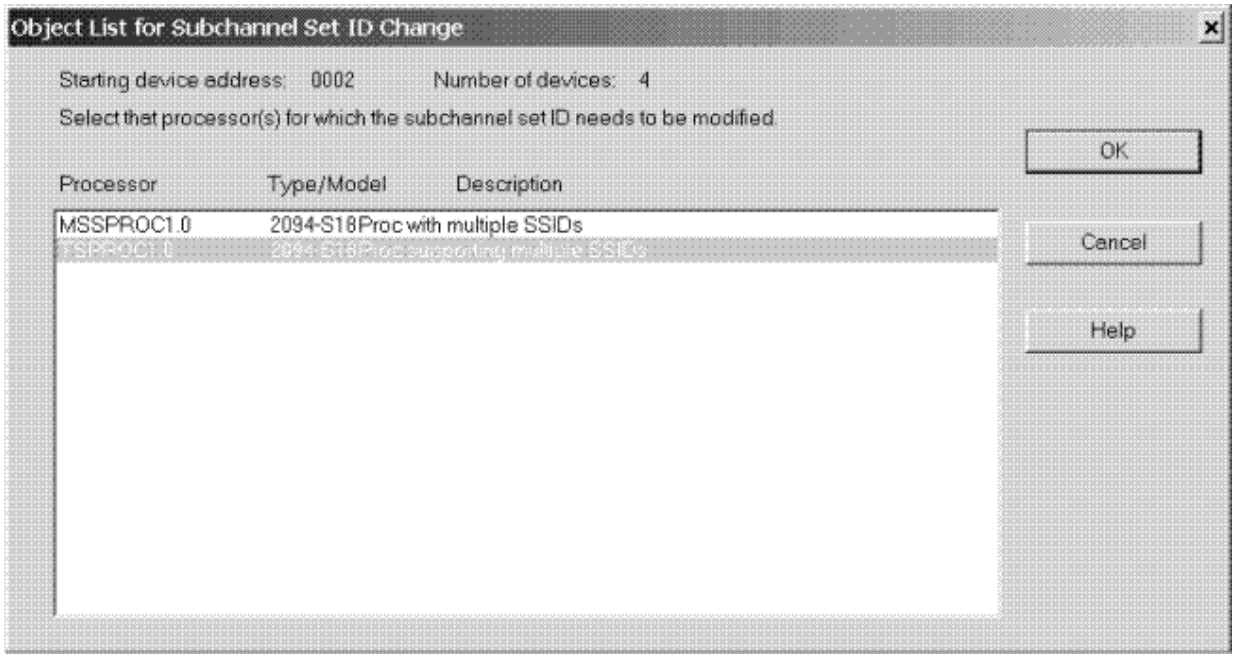


Figure 83. Selecting processors for changing the subchannel set placement for devices

2. In the subsequent **Edit Subchannel Set ID** dialog for processors, you set the subchannel set ID for the device <-> processor relation.

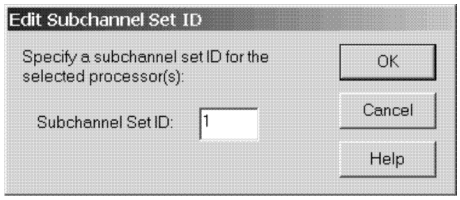


Figure 84. Edit Subchannel Set ID

3. In the **Object List for Subchannel Set ID Change** dialog for operating systems, you select the operating system(s) for which you want to change the subchannel set ID for the device <-> operating system definition. This step is similar to step “1” on page 149.
4. In the subsequent **Edit Subchannel Set ID** dialog for operating systems, you now set the subchannel set ID for the device <-> operating system relation. This step is similar to step “3” on page 150.

Modifying the subchannel set ID for 'Device <-> Operating System' only:

Start from the menu bar choice **Edit** and select **OS configurations...**. Then select one operating system from the **Operating System Configurations** dialog and press the **I/O Devices...** button. This invokes the **List Devices for OS Configuration** where you select a single device or a group of devices and press either the **Edit...** button or the **Group Edit...** button. This leads you to the **Define Device <==> OS Configuration Parameters** dialog where you can modify the subchannel set ID.

Modifying the subchannel set ID for 'Device <-> Processor Definitions' only:

From an **I/O Device List** select one or more devices and press the **Dev <=>Proc ...** button. From the **I/O Device <=> Processor Definitions** dialog select one or more processors / channel subsystems and press either the **Edit...** button or the **Group Edit...** button to invoke the **Edit I/O Device <==> Processor Definitions** dialog where you can modify the subchannel set ID.

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.

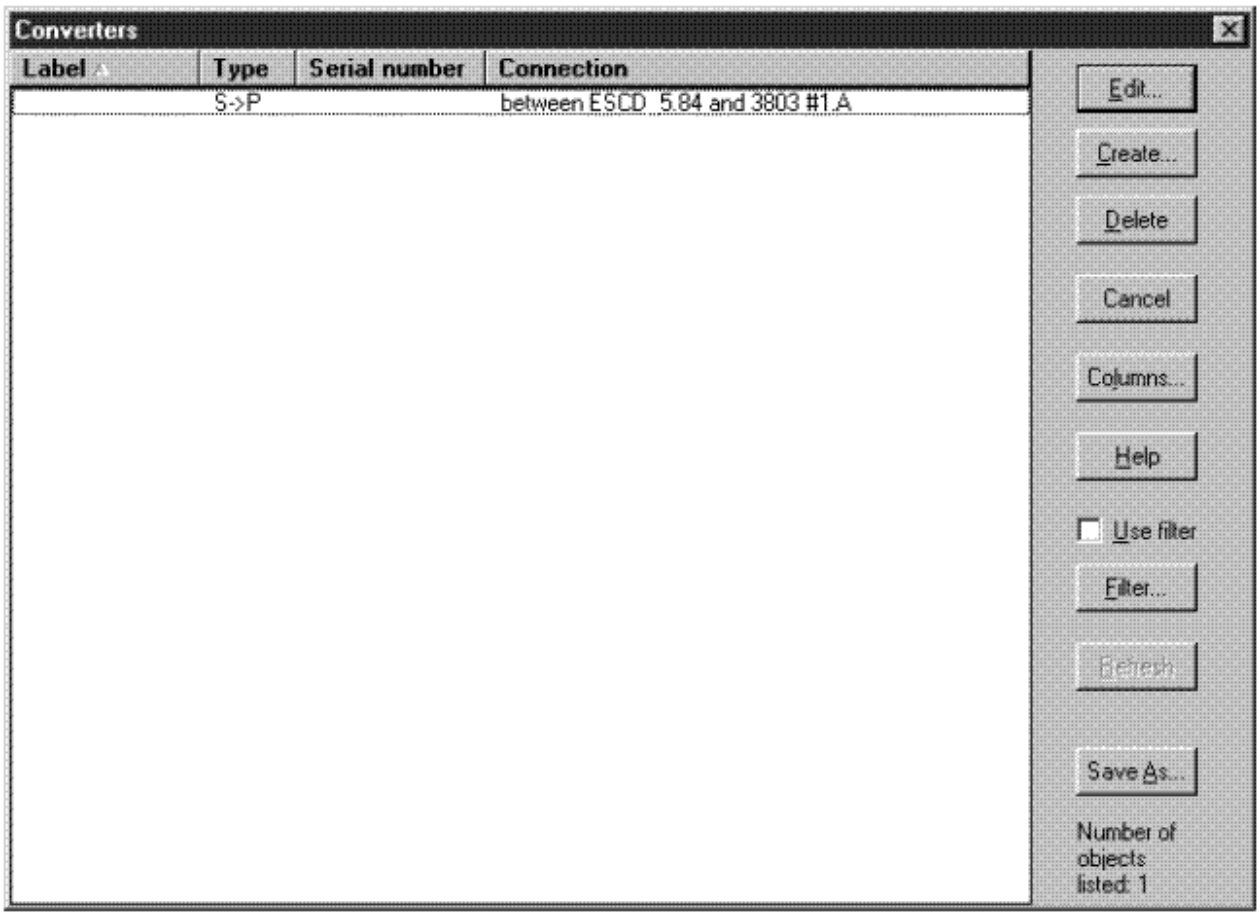


Figure 85. Converters

Creating a converter

In the **Converters** dialog, press **Create** to open the **Create Converter** dialog.

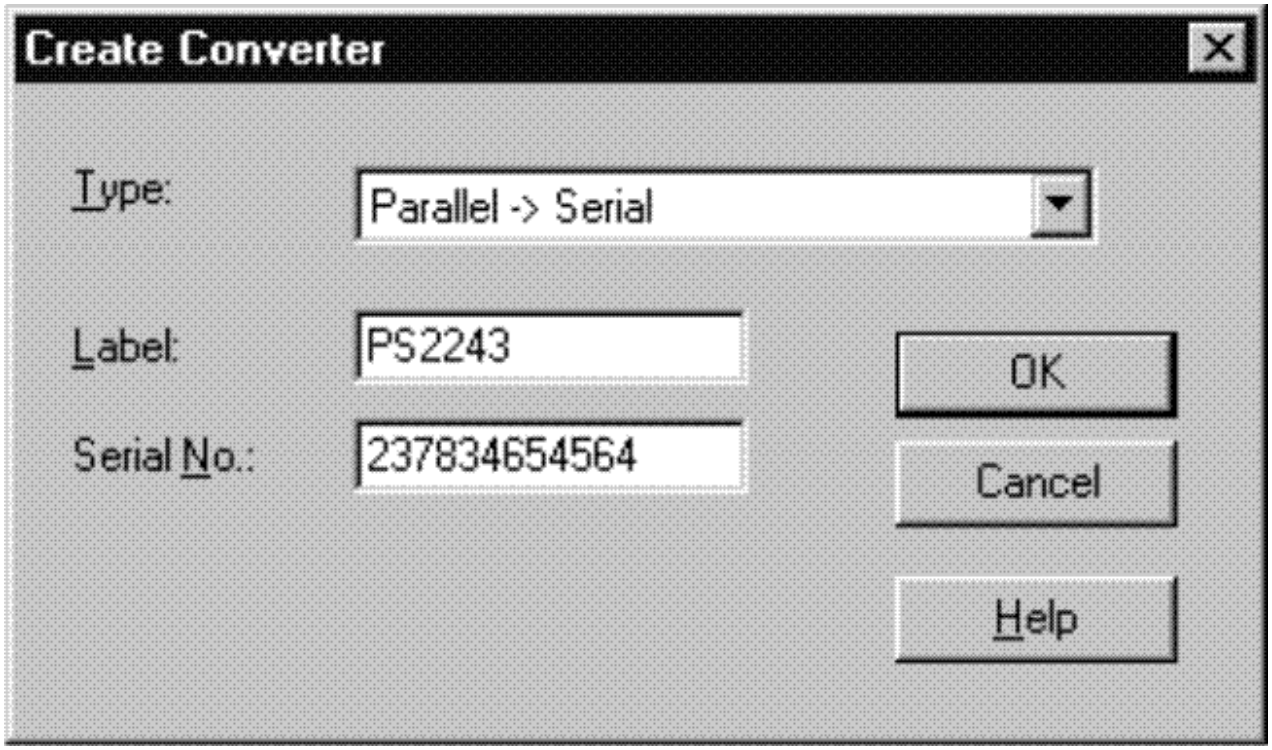


Figure 86. Create Converter

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 212.

You can replace this converter with an unused converter or a newly created one by pressing the **Replace...** button.

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.

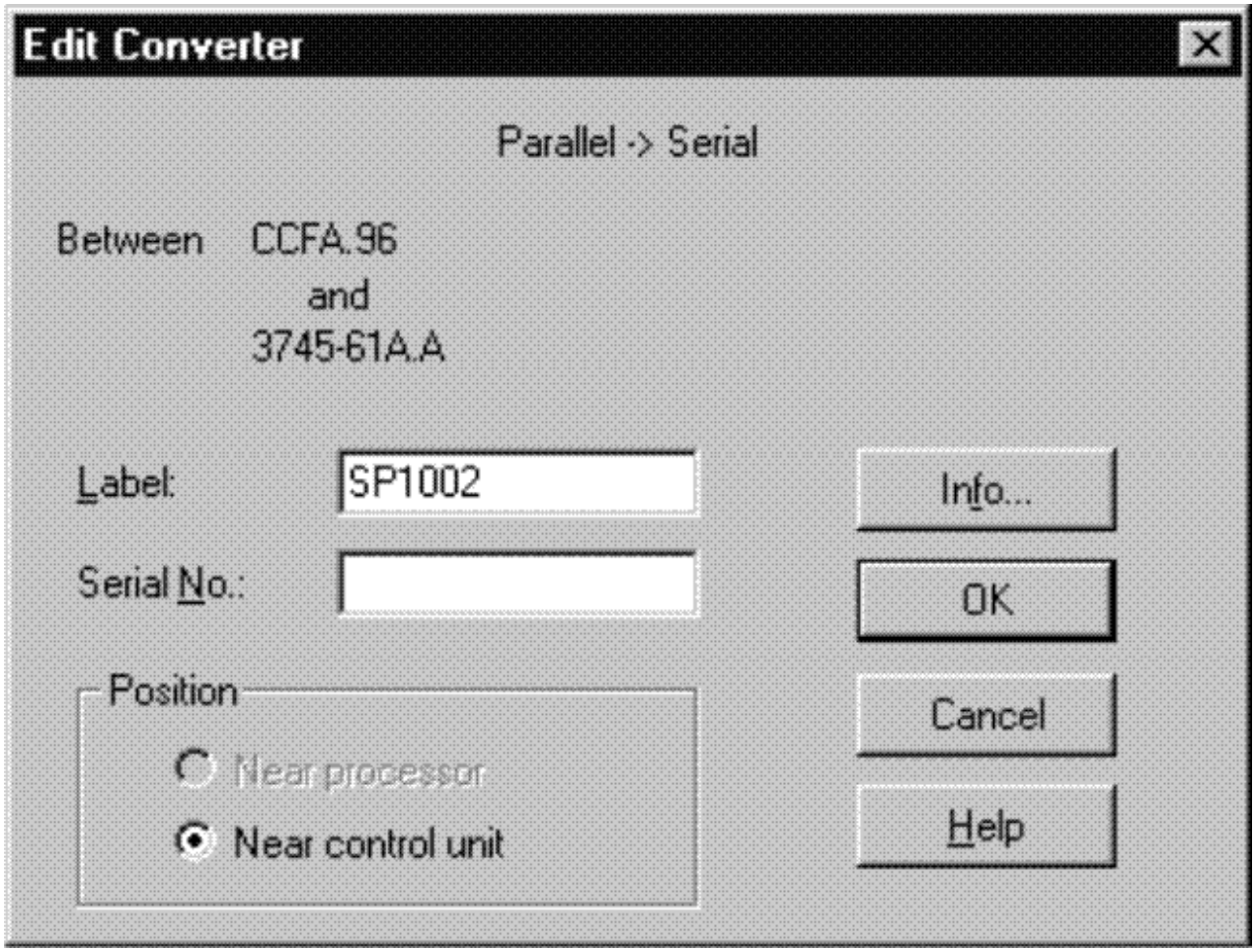


Figure 87. Edit Converter

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.

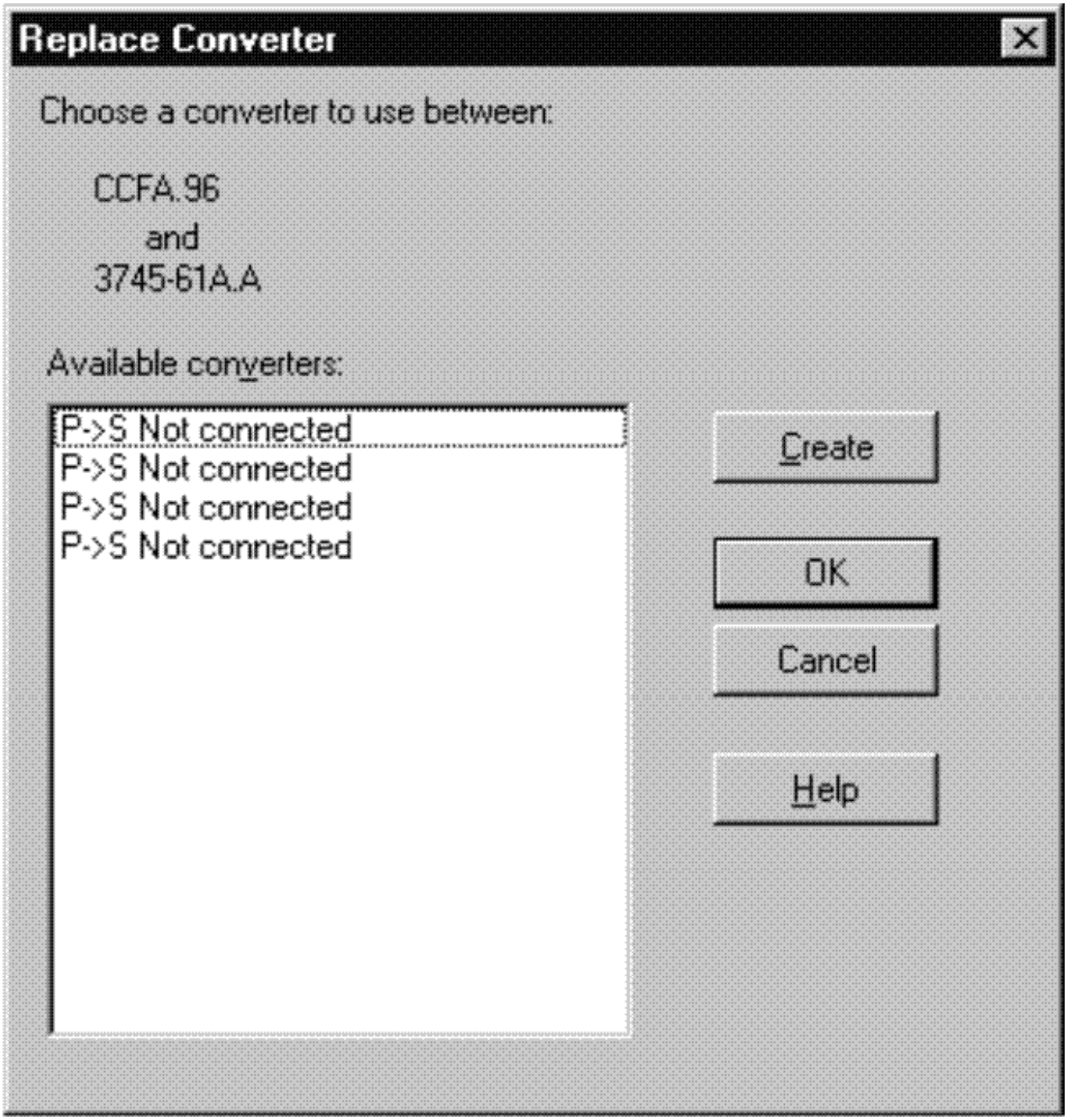


Figure 88. Replace Converter

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 151).

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 38 for information about 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 175 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.

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 156.

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 89 on page 156 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.

Note: Pressing the **Save Port List...** button lets you save the information from the **Ports** list into a text file.

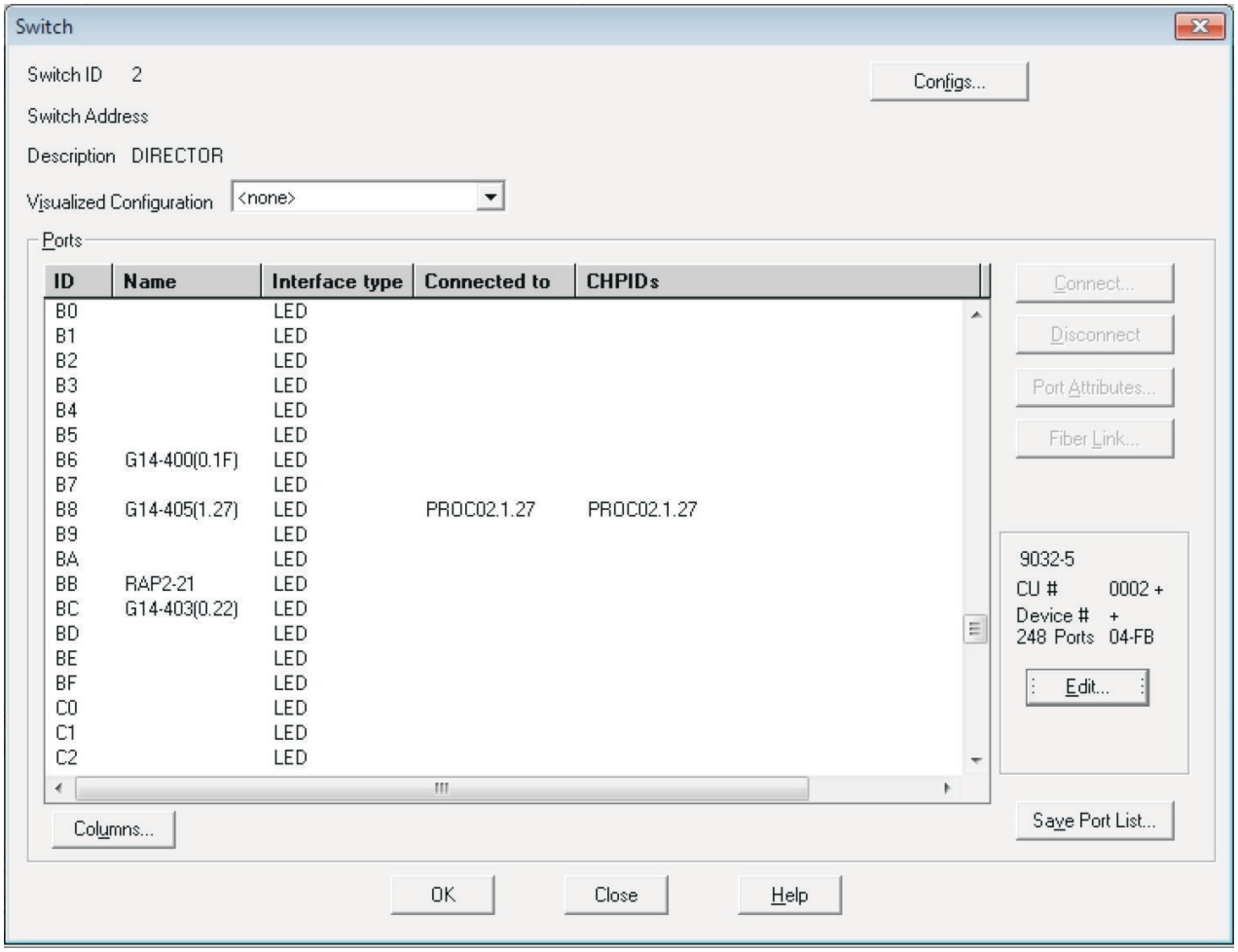


Figure 89. Switch

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.

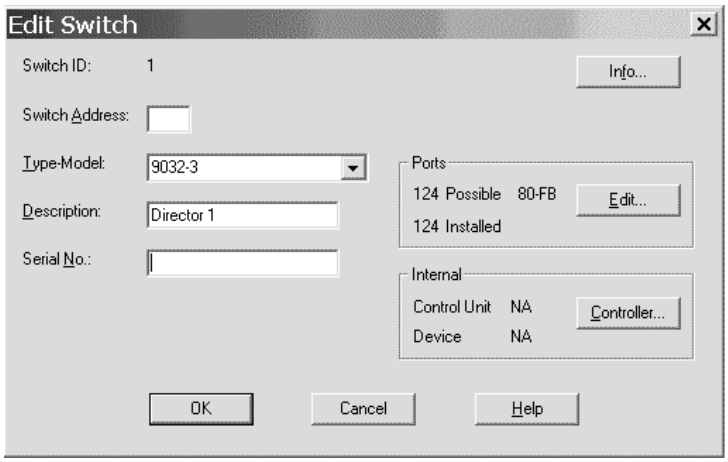


Figure 90. Edit Switch

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 29.

In the **Ports** section, you see how many ports are installed from the possible range. To change the installed ports or port ranges of the switch, 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 be removed (uninstalled), or, if not yet all ports are installed from the possible range, you can extend the switch's installed port range by installing new ports. HCM assigns to 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 130 and “I/O devices” on page 146 for more details.

As in the **Switch** dialog in Figure 89 on page 156 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.

Editing switch port attributes

In the **Switch** dialog, select one or more switch ports from the list box and click on the **Port Attributes...** button. 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.

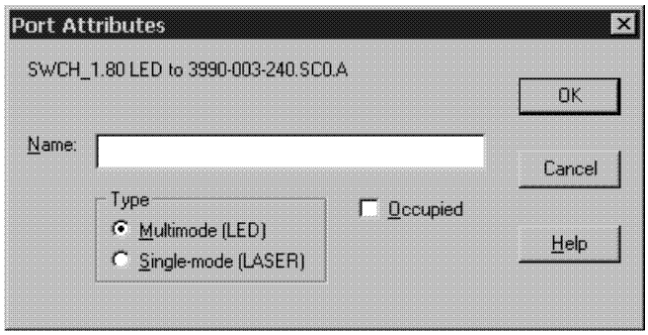


Figure 91. Port Attributes

Applying default port names

You can automatically apply standard names to all ports of one or more switches. Select **Switch...** from the **Edit** menu bar choice. The **Edit a Switch** dialog appears, showing the list of defined switches. Select one or more switches for which you want to generate default switch port names and press the **Port Names** button. This action composes a default name for all involved ports, taking up to the first 24 characters of the corresponding connection information. This connection information can also be seen from the ***Connected to*** column of the ***Switch*** dialog, when accessing the related switch from the diagram. HCM issues a message that lets you decide whether you want to accept the composed default names.

Note: There is another possibility to apply such default port names to a single switch when you are working with that switch: see “Applying switch port names” on page 161.

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 212 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 231 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 140.

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. For z/OS, you can not work with the configurations stored on the switch with HCM releases higher than V2R3.

In previous releases, HCM used the I/O Ops component for retrieving, manipulation, and activating switch matrices. I/O Ops is withdrawn and is no longer supported with HCM and HCD releases higher than V2R3. Withdrawing the I/O Ops support in HCM means that HCM no longer supports retrieving, manipulation, and activating switch matrices.

Though it is still possible to define switches and switch configurations in HCM, the switch information in IODF can become out of sync with information in the switch itself. Therefore, it is recommended that you stop using the switch configuration information in HCM.

Note: More recent FICON switches are configured within the switches themselves.

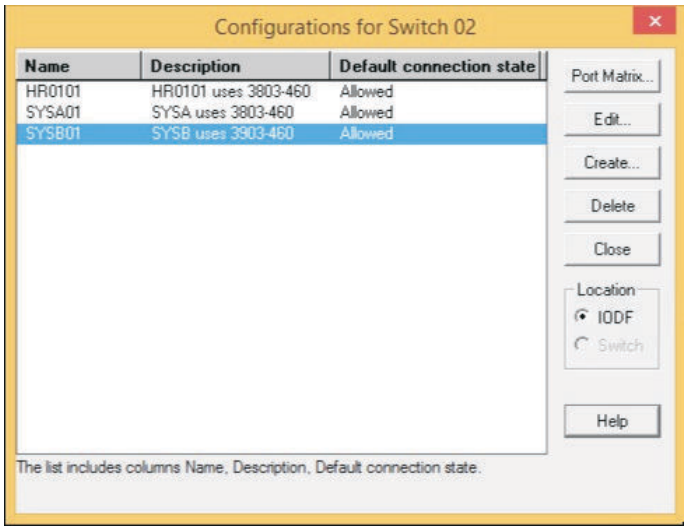


Figure 92. Configuration for Switch XX

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 are analyzed by HCD and in the case of chained connections, all possible paths of chaining switches are 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.

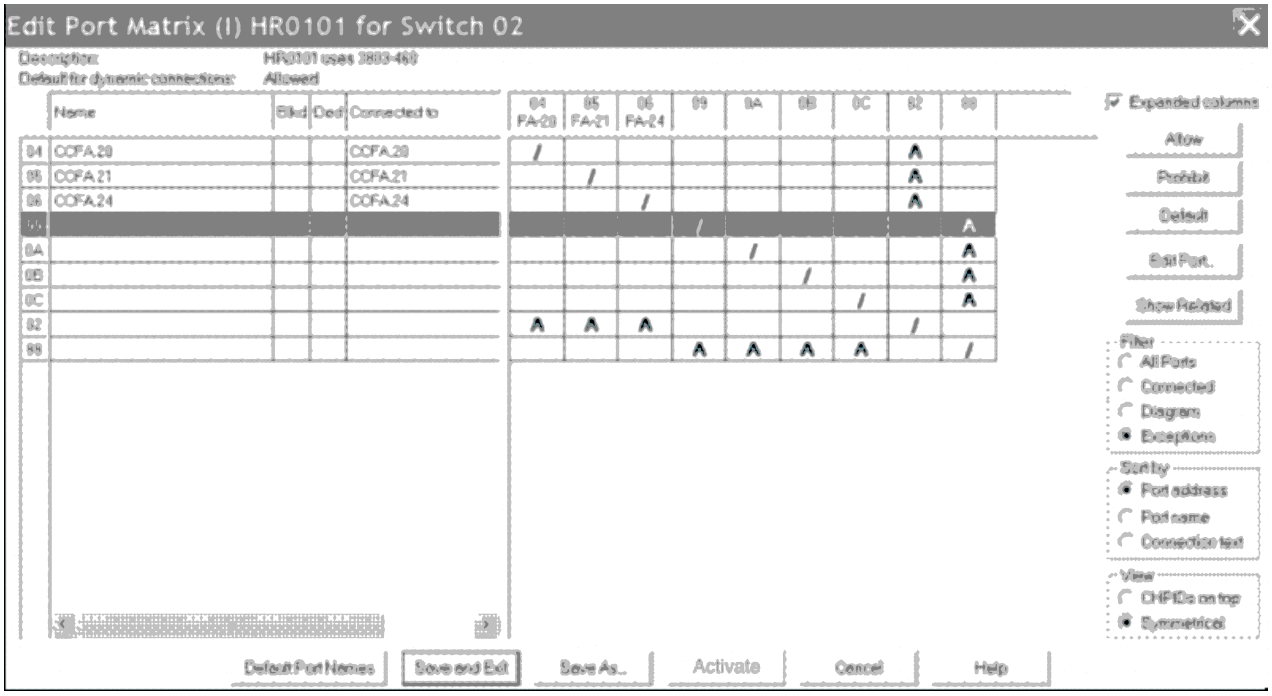


Figure 93. Configuration Port Matrix

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 12. Symbols in port matrix

Symbol	Description
Black square	These ports have a dedicated link to each other.

Table 12. Symbols in port matrix (continued)	
Symbol	**Description**
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.
Slash (/)	For ESCON port matrices, you cannot specify a value (default, allow, prohibit) for the diagonal of the port matrix which handles connections from a port to itself. Hence, the matrix contains a '/' on the diagonal indicating that the switch matrix diagonal cannot be edited.

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.

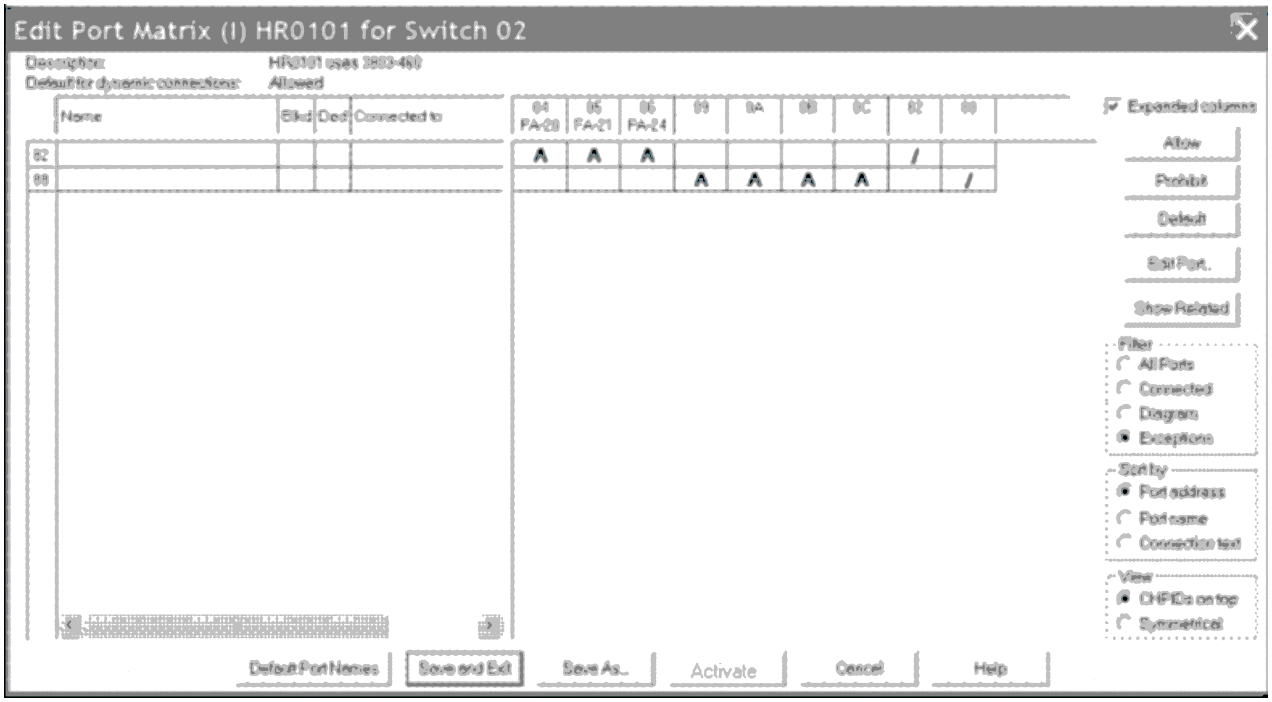


Figure 94. Configuration Port Matrix - Filtered and CHPIDs on Top

Editing port-to-port connections

For ESCON switches, you cannot edit the diagonal entries of the port matrix; that is you cannot define a connection from a switch port to itself. However , for FICON switches only, you can set the matrix entry on the diagonal explicitly to the values **Allow** or **Prohibit** for a port connection to itself (loopback connection).

For switches other than FICON, in the ***Edit Port Matrix xxx for Director yyy*** dialog, the matrix contains a '/' on the diagonal indicating that the switch matrix diagonal cannot be edited . If you attempt to update the diagonal of an ESCON port matrix, HCM shows a message that this action is not allowed.

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 38 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.

Applying switch port names

There are two possibilities to apply names to switch ports: you can apply a default name to all connected ports in one step or you can edit names of single ports.

To edit a single port name, select a port from the ***Edit Port Matrix*** dialog (Figure 93 on page 159) and press the ***Edit Port...*** button. This invokes the ***Edit Switch Port...*** dialog which lets you specify a name.

To automatically apply standard names to all ports, press the ***Default Port Names*** button from the ***Edit Port Matrix*** dialog. HCM then composes a default name for each connected port. This name is composed from the port 's connection information (shown in the ***Connected to*** column). If no connection information is available, no name is applied. You can afterwards change the default name for single ports as described above .

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 is no longer supported. In previous releases, HCM used the I/O Ops component for activating switch configurations. But I/O Ops is withdrawn and is no longer supported with HCM and HCD releases higher than V2R3.

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 175 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.

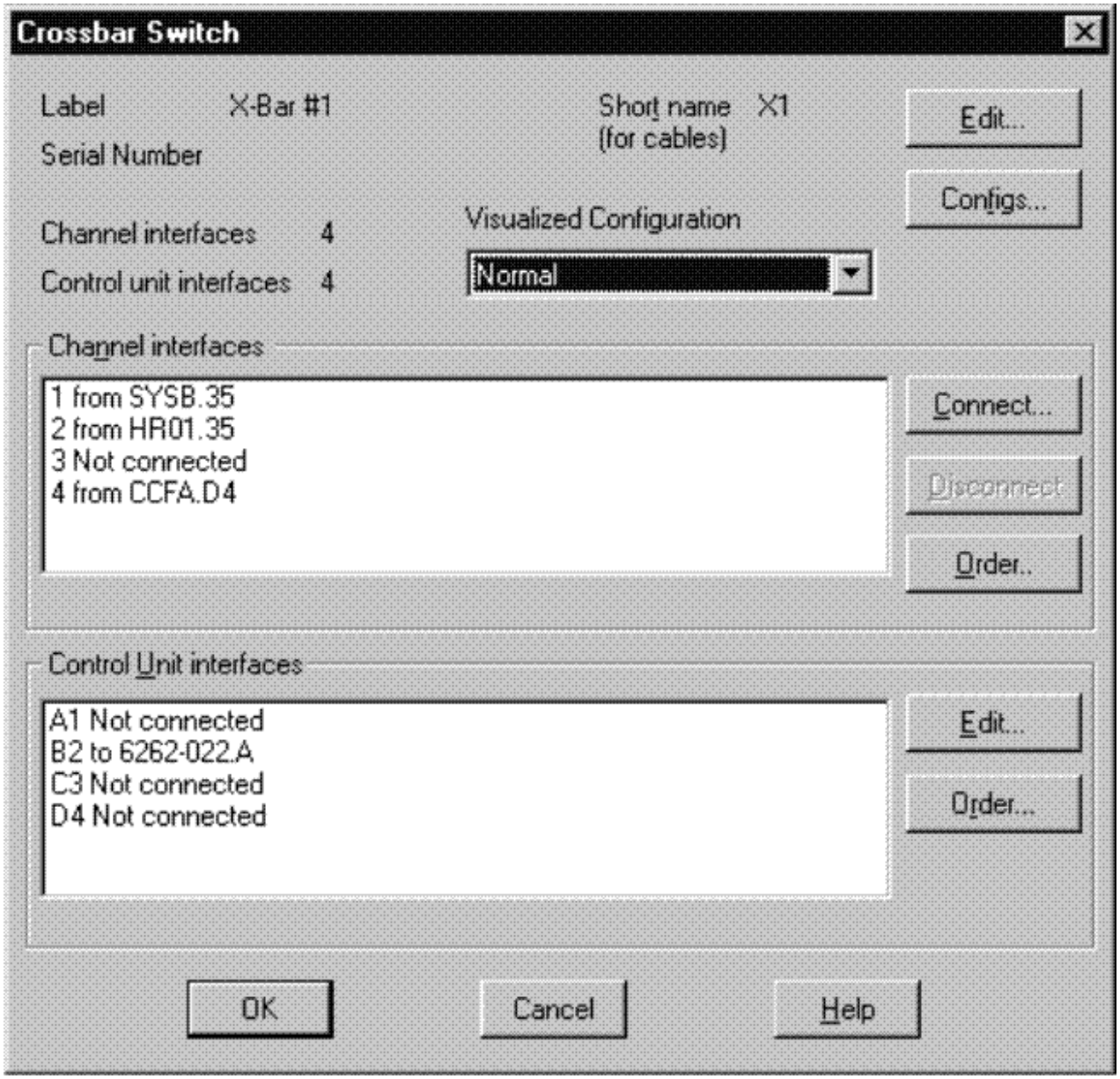


Figure 95. Edit Crossbar Switch

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 29.

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.

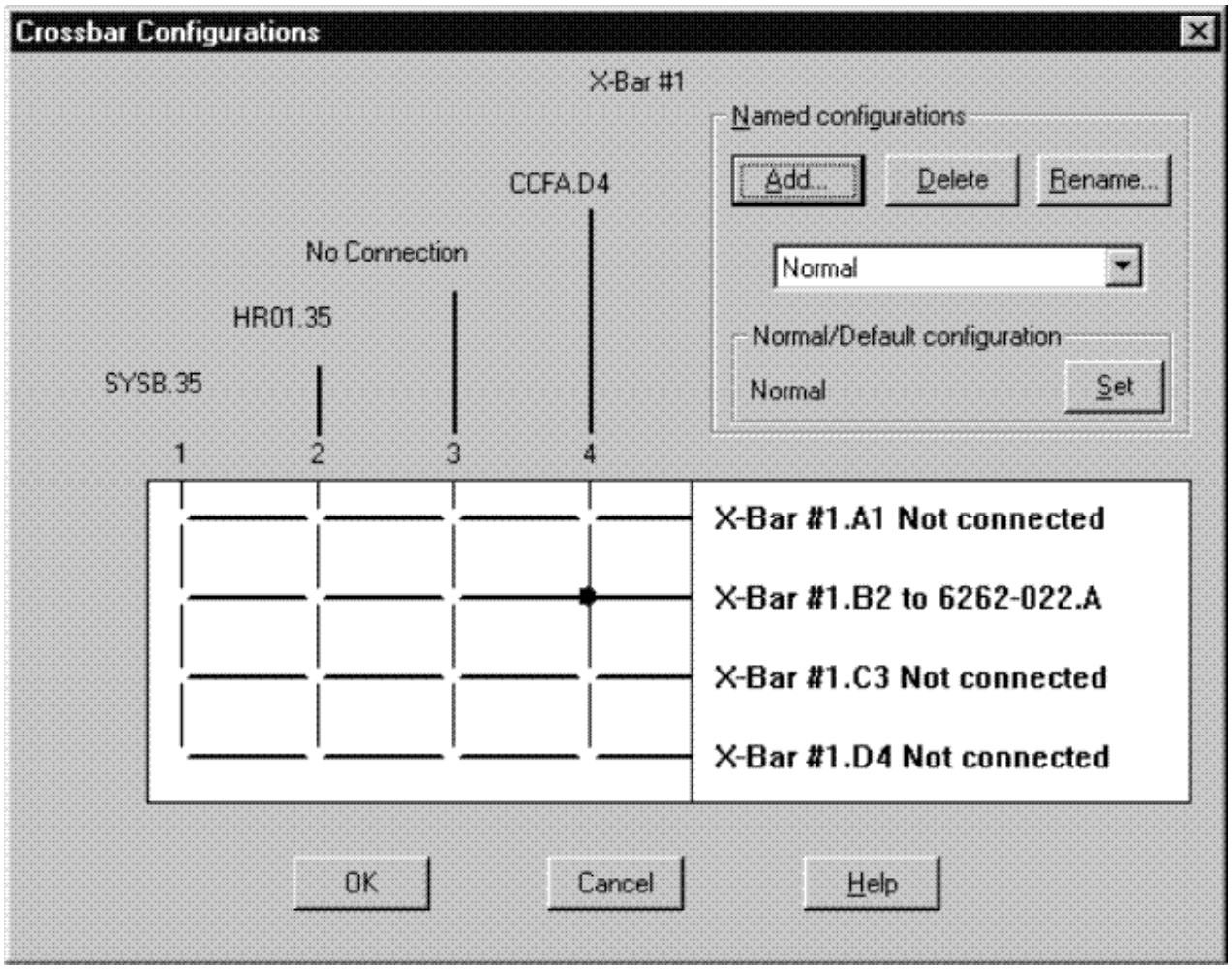


Figure 96. Crossbar Configurations

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 46. 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 38 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 212 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/general box port naming format” on page 170 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 212 and “Infrastructure connections” on page 212.

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 167). 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 29.

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 97 on page 168). 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/general box port naming format” on page 170.

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 97 on page 168.

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.

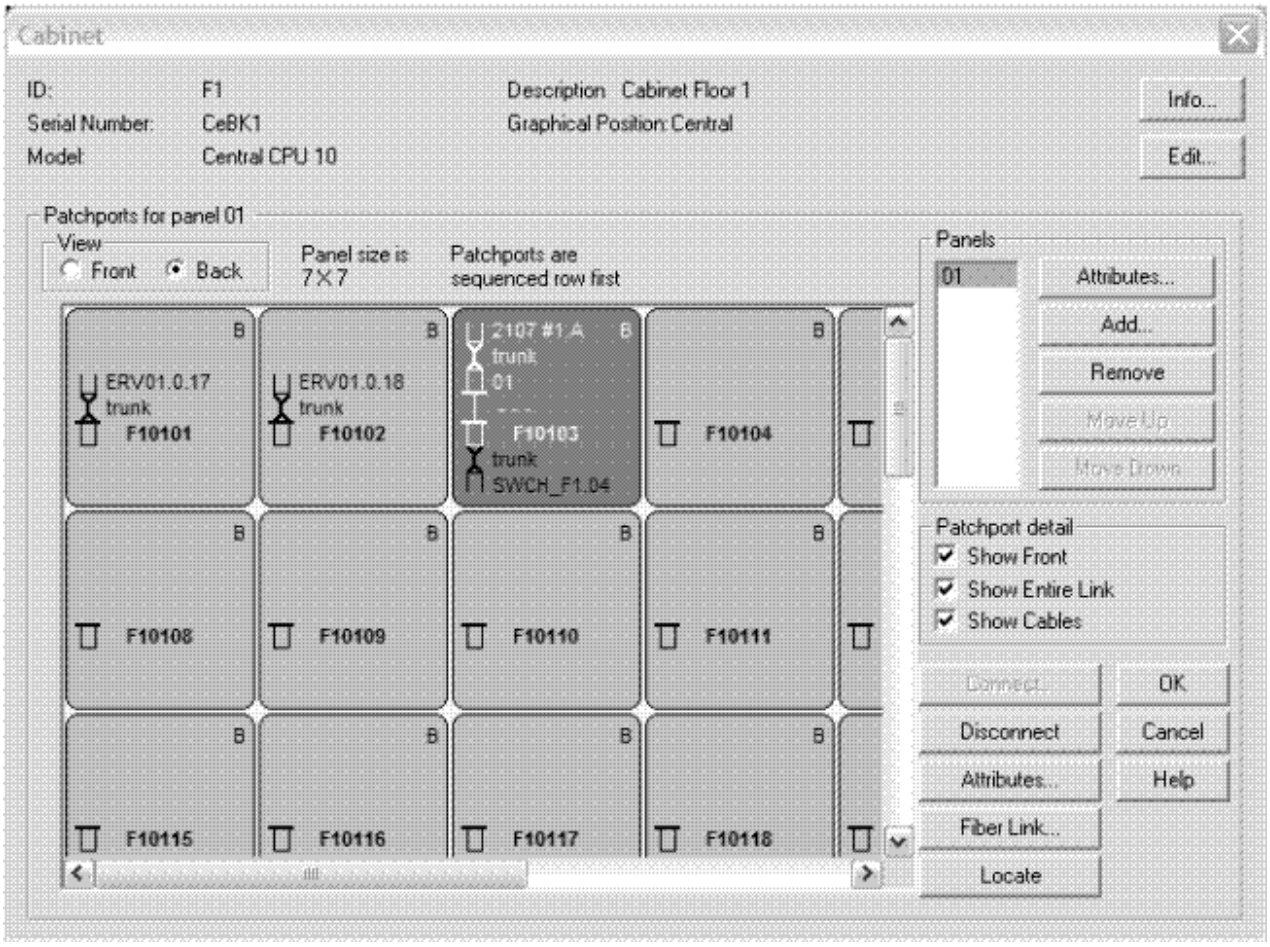


Figure 97. Cabinet

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 center 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 causes 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 are 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 (HCM asks you 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 212 for information about 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/general box port naming format

To create user-defined patchport/general box port 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 port name is displayed.

Following proxies are allowed:

%CABINET% or %NAME%
ID of the cabinet or name of the general box. Both proxies are synonyms and can be used for either cabinets or general boxes.

%PANEL%
panel number

%GROUP%
port group number

%PORT%
patchport/general box port number

%ROW%
the row within the panel containing the port

%COLUMN%
the column within the panel containing the port

The numerical proxies (%PANEL%, %GROUP%, %PORT%, %ROW% and %COLUMN%) 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 are used to pad to the appropriate width; if the conversion is alphabetical, leading underscore characters are used. If the width is zero or omitted, the actual text uses 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>
CABINET, NAME, PANEL, GROUP, PORT, ROW or COLUMN

<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.

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 backslash '\' 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 backslash.

General boxes

You can create general boxes for purposes that are not covered by other HCM objects described in the previous sections of this chapter. Such general boxes can represent physical devices and their connections in your configuration that are not part of the logical definitions of an I/O configuration. For example, you can use general boxes for network devices or non-zSeries boxes and their connections. These objects or connections are stored in the HCM configuration file, but not in the associated IODF. The general boxes appear in the configuration diagram, so that you can have a complete view of the physical objects. Another purpose of general objects is to document the complete cabling of your devices in your configuration.

A general box is a container for resources. General boxes are organized like cabinets: one general box can contain several panels, and each panel can contain general box ports, ordered in rows and columns. You can organize your resources with general boxes according to your needs.

Creating a general box

Invoke the **Create General Box** dialog (Figure 98 on page 172) from the **Create** menu.

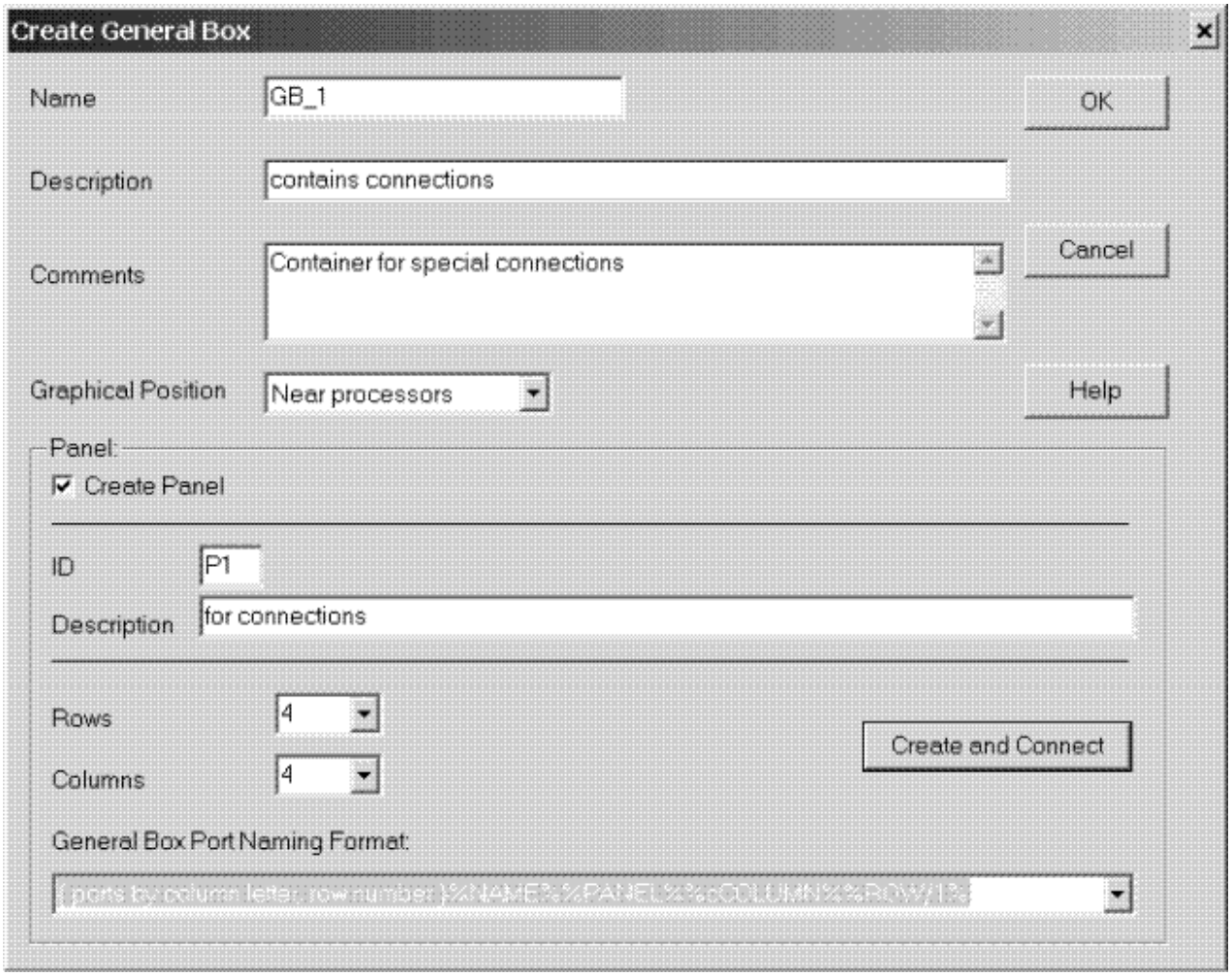


Figure 98. Create General Box

Enter the required information for the general box. In the **Panel** box, when checking the **Create Panel** check box, you can select to create the general box together with a first panel. Pressing **OK** just creates the general box, while pressing **Create and Connect** directly leads you to the **General Box** dialog where you can specify the connections of your objects (see **General Box** dialog in "Editing a general box" on page 172).

Editing a general box

Use the following dialogs to edit the properties of a general box:

- The **Edit General Box** dialog is for editing certain box attributes, for example the name, description and graphical position. For this purpose, use the **Edit General Box** dialog, which you invoke using **Edit...** from the context menu of the general box.
- The **General Box** dialog is for editing the connectivity contents, for example to add new panels or connect the general box ports. For this purpose, double-click on the general box in the configuration diagram. Alternatively, from the ***Edit*** menu bar choice, select ***General Box...***, then select a general box from the list and press **OK**.

  In both cases, you reach the **General Box** dialog where you can specify the connections of your objects. This dialog looks similar to the **Cabinet** dialog shown in Figure 97 on page 168. Y ou can work with general box ports in the **General Box** dialog just like you work with patchports in the **Cabinet** dialog as described in "Patchports" on page 167.

  Especially, you can link two general box ports together using the **Connect** button in the **General Box** dialog and then assign a cable to that link as described in "Managing physical cables" on page 207.

Deleting a general box

If you want to delete a general box, you must disconnect the used general box ports first using the **Disconnect** button in the **General Box** dialog. Then select the object in the configuration diagram and use the **Delete** choice from its context menu or press the DEL key.

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.

© Copyright IBM Corp. 1995, 2021

- *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 183.

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 207 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.

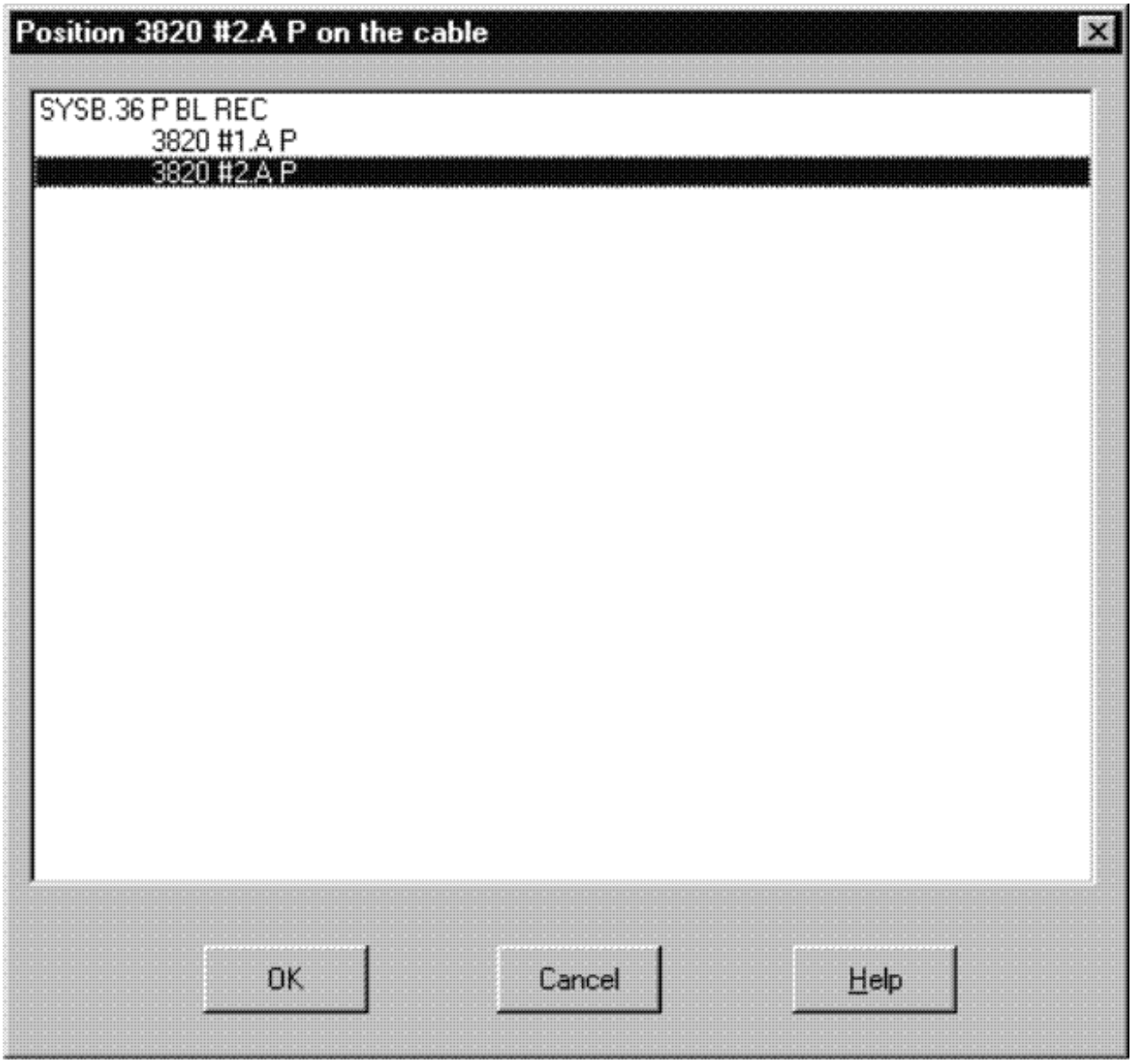


Figure 99. Positioning an Object for Daisy-Chained Connections

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.

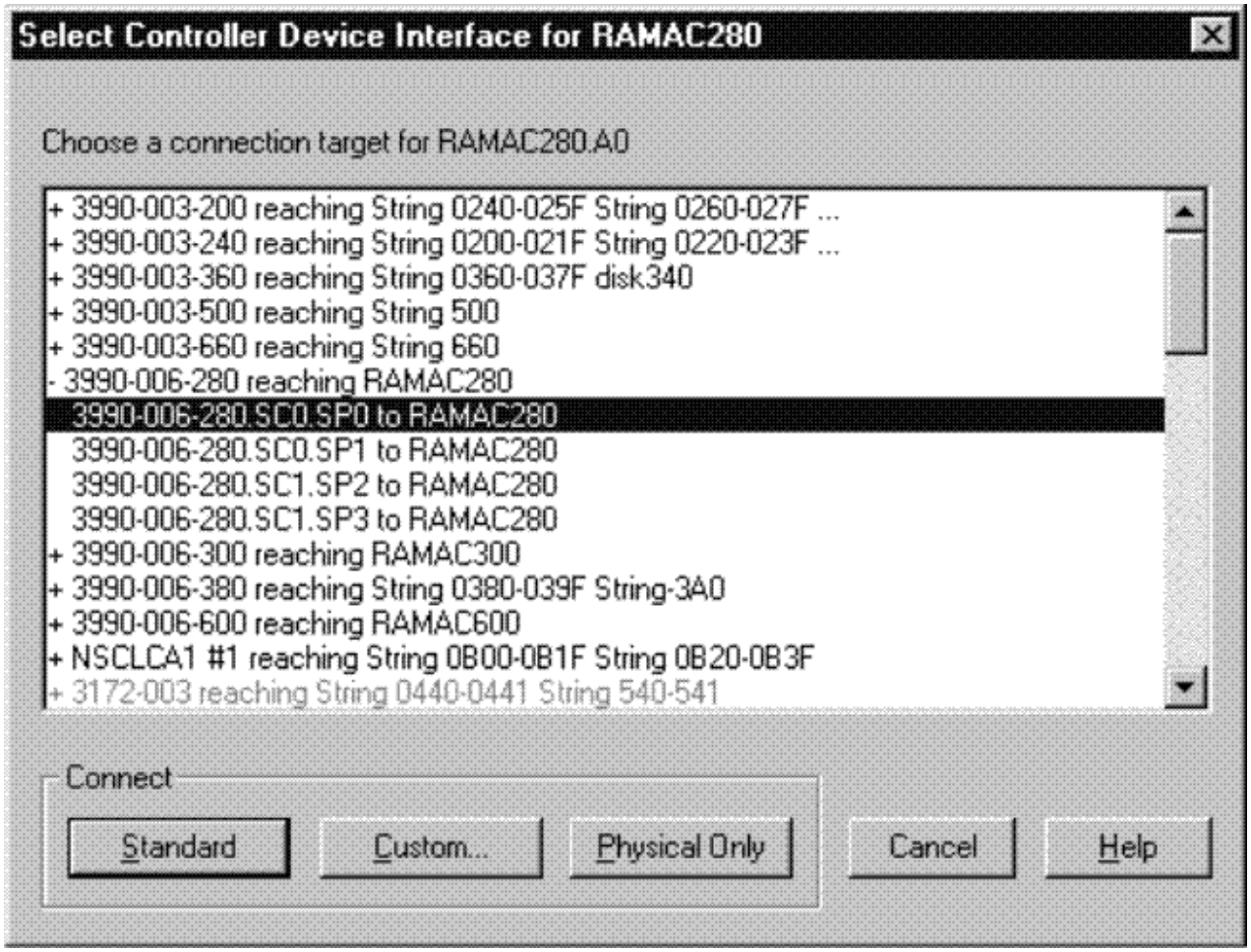


Figure 100. Select Controller Device Interface

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 178), 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.

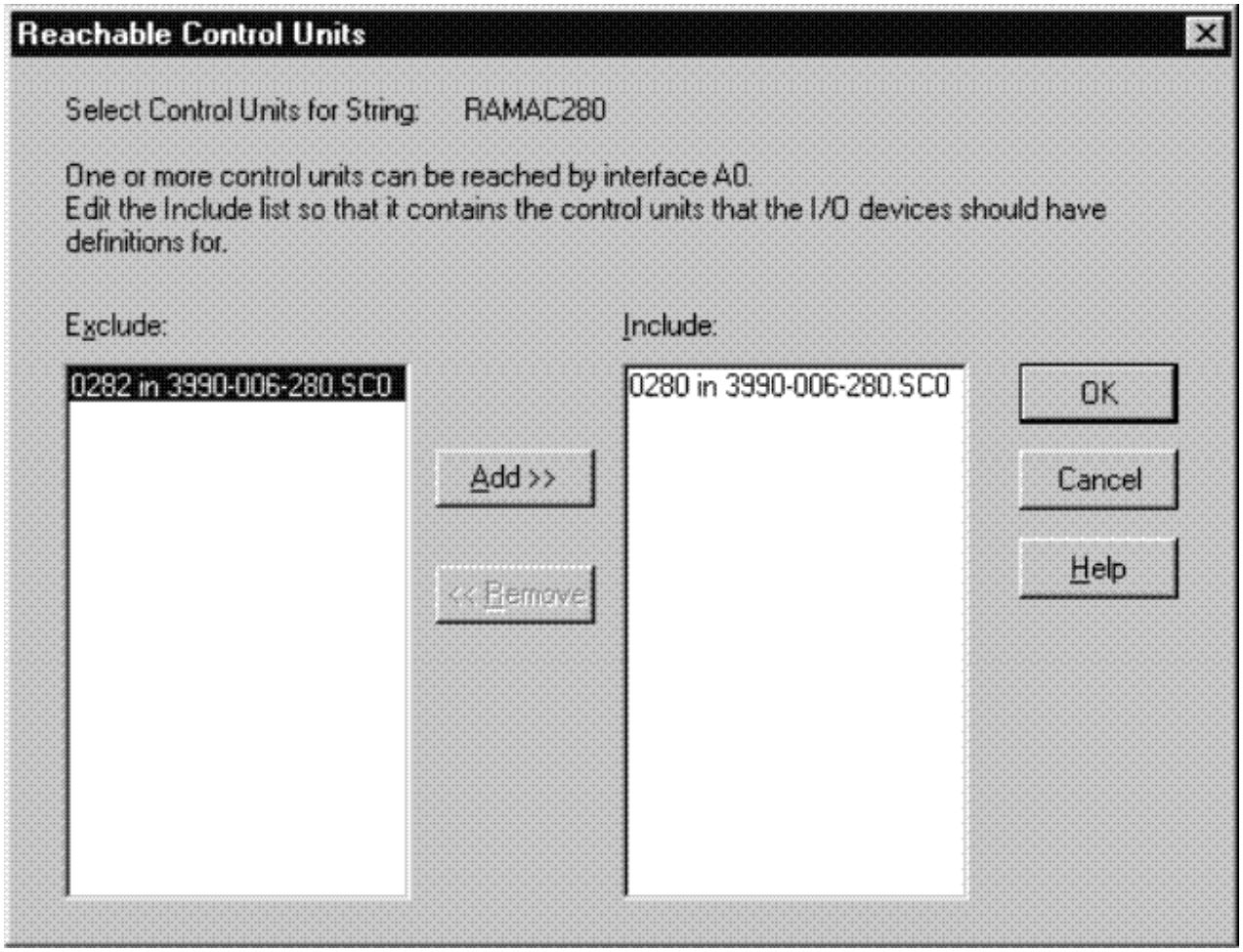


Figure 101. Reachable Control Units

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.

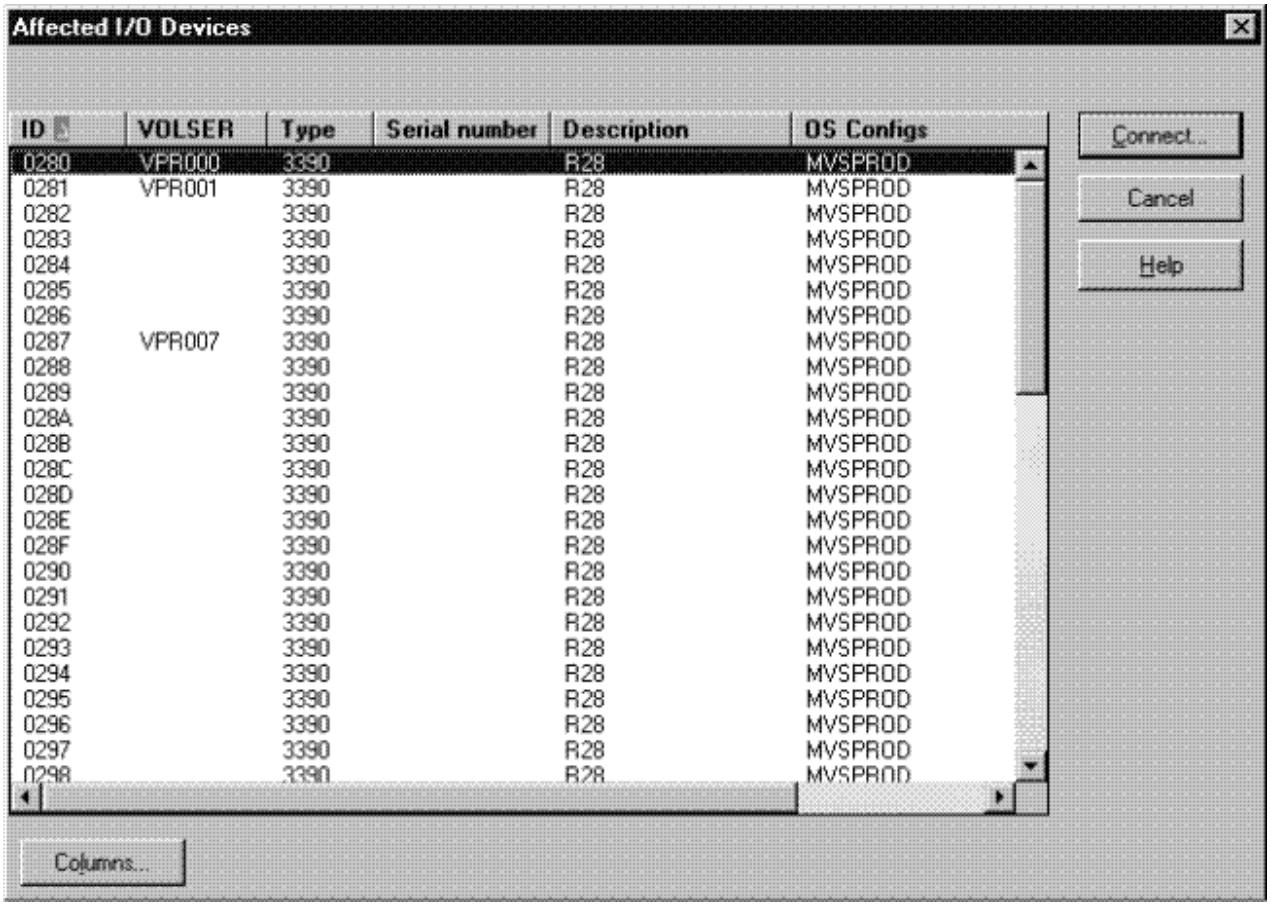


Figure 102. Affected I/O Devices

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.

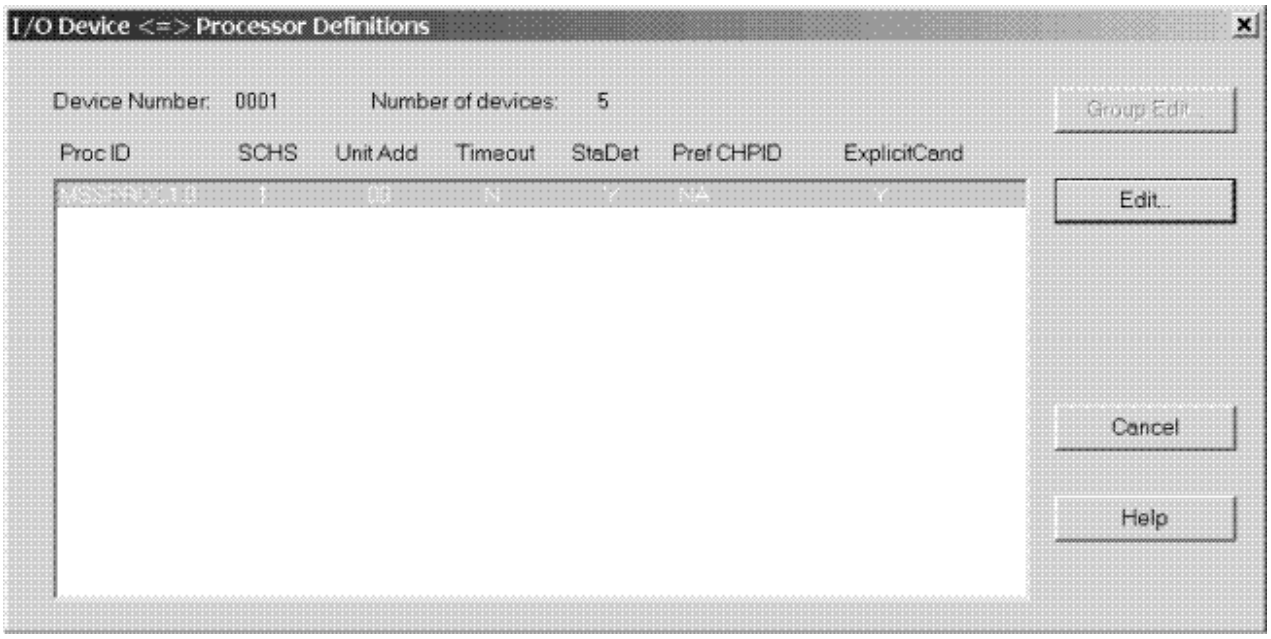


Figure 103. I/O Device <=> Processor Definitions

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.

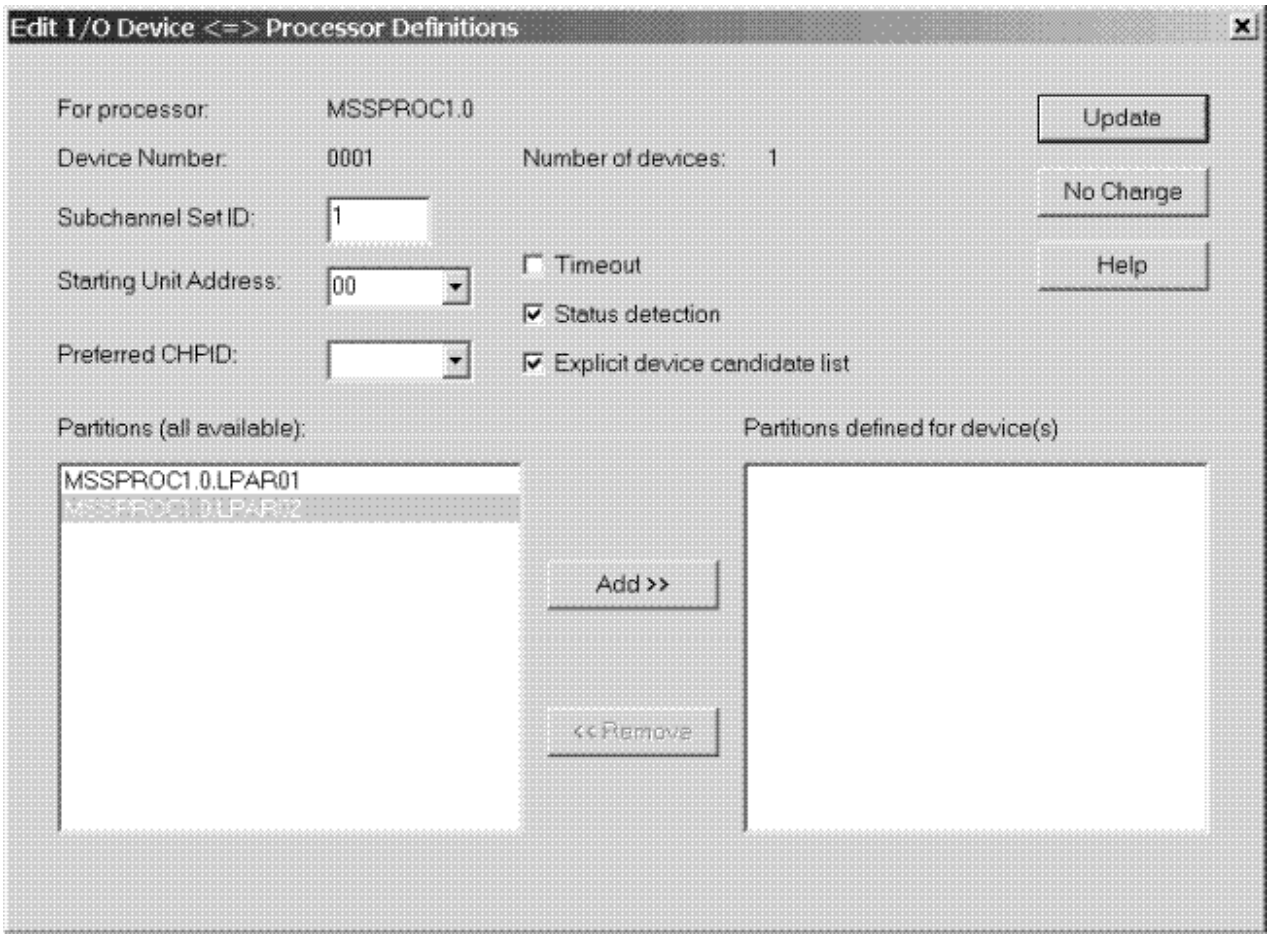


Figure 104. Edit I/O Device <=> Processor Definitions

In this dialog, you can modify the attributes of the logical definition for the selected device(s), control unit and processor(s). The default subchannel set is 0. You can change this subchannel set only for PAV alias devices attached to processors supporting multiple subchannel sets.

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 104 on page 182, 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 212 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 about how to use this dialog, refer to “Connecting switches to processors or other switches” on page 193.

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 218 and “Connecting switches to processors or other switches” on page 193 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.

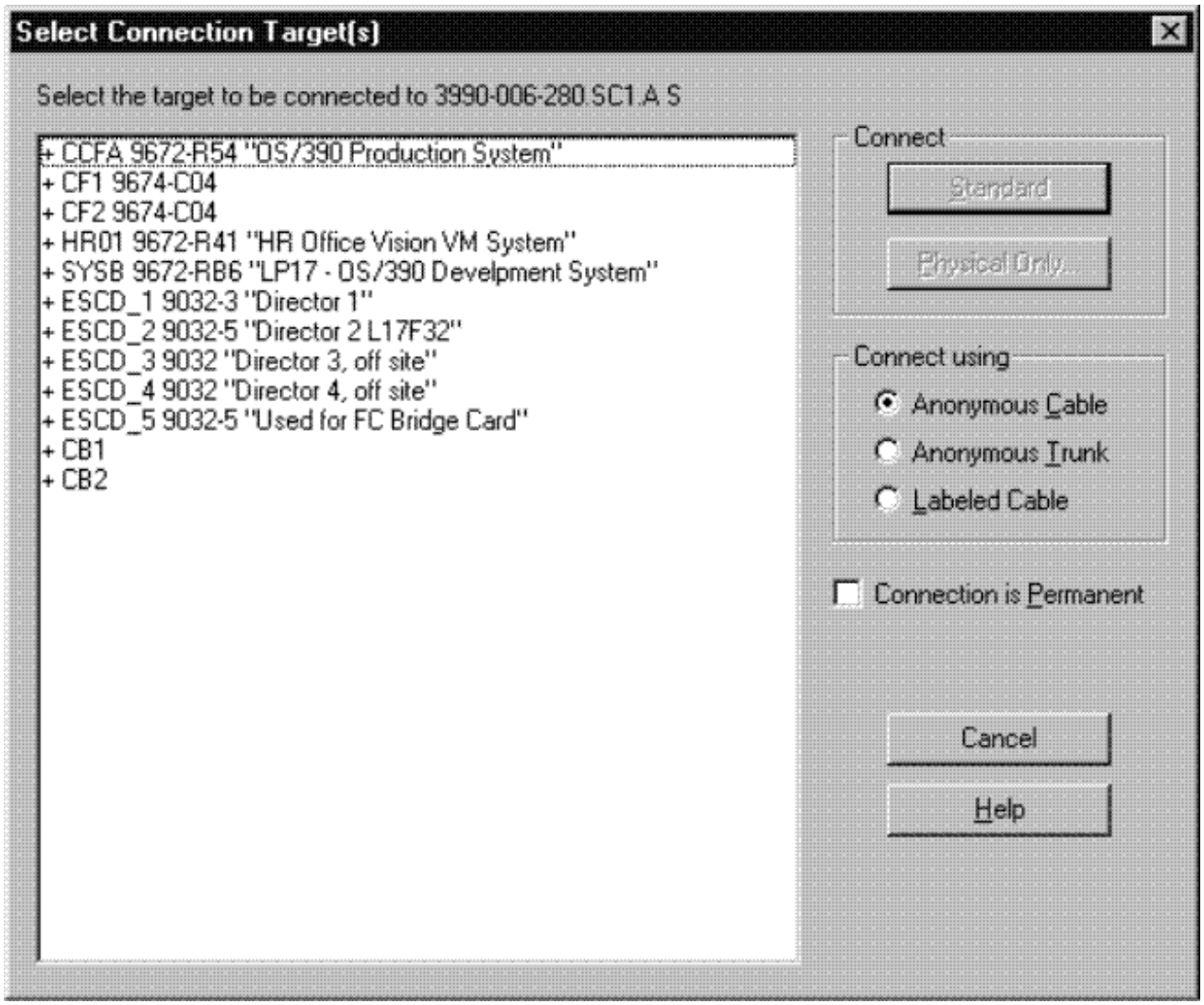


Figure 105. Select Connection Target(s)

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 176 for more information about 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 42 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
- **Choose Channel Subsystems**

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.

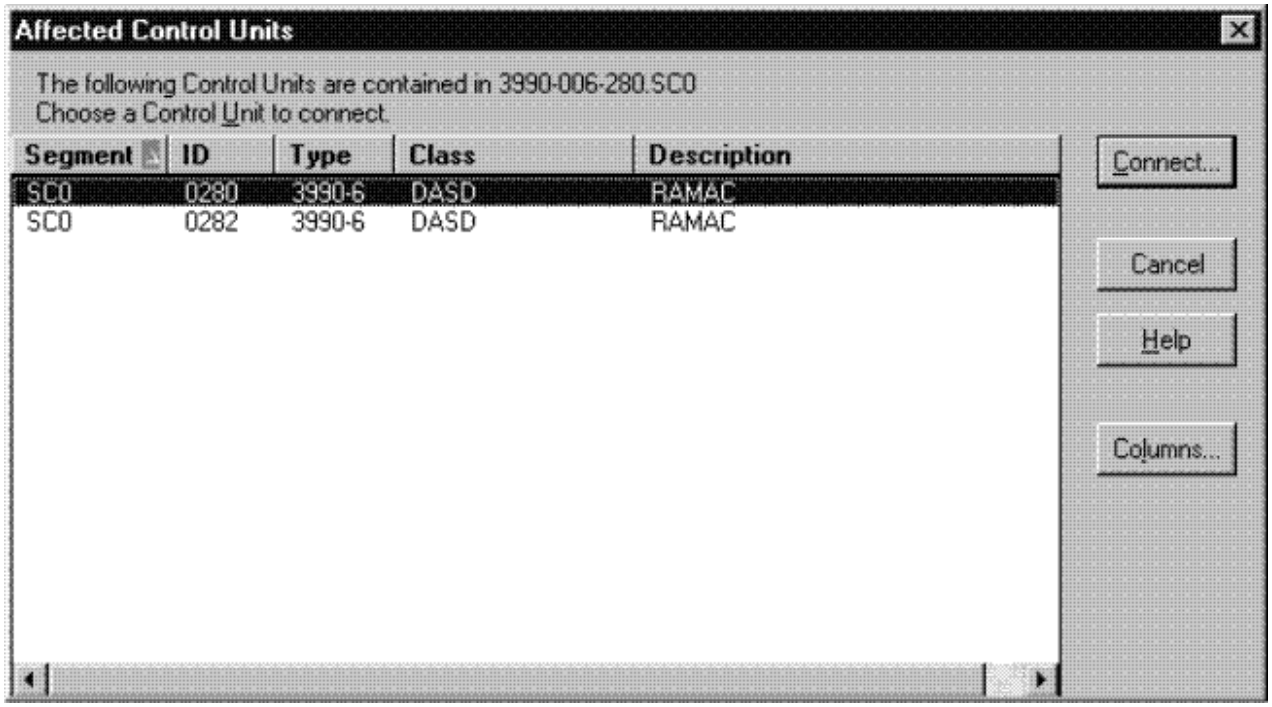


Figure 106. Affected Control Units

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.

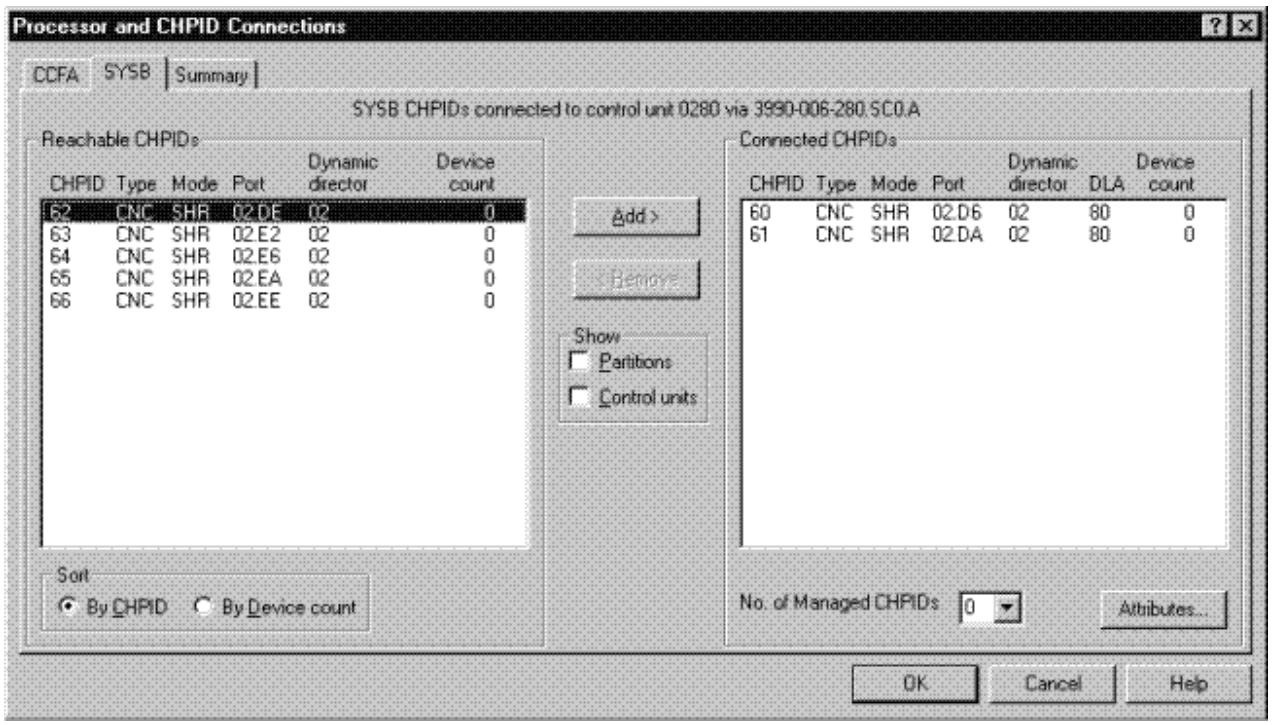


Figure 107. Processor and CHPID Connections

The **Processor and CHPID Connections** dialog appears if you connect the control unit via a switch.

If you selected multiple control units that do not have the same static CHPID connections, HCM asks for your permission to proceed: the dialog presents only the connections of the ***first*** selected control unit.

Changes will, however, affect ***all*** selected control units, by establishing exactly the shown connections (which may break existing CHPID connections that are ***not*** shown in the dialog).

You can select the available channel paths of a processor by clicking the appropriate processor tab in the dialog. The summary tab in this dialog gives an overview of which CHPIDs the connection is established. 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 shows 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 108 on page 188.

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 creates 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 creates 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.

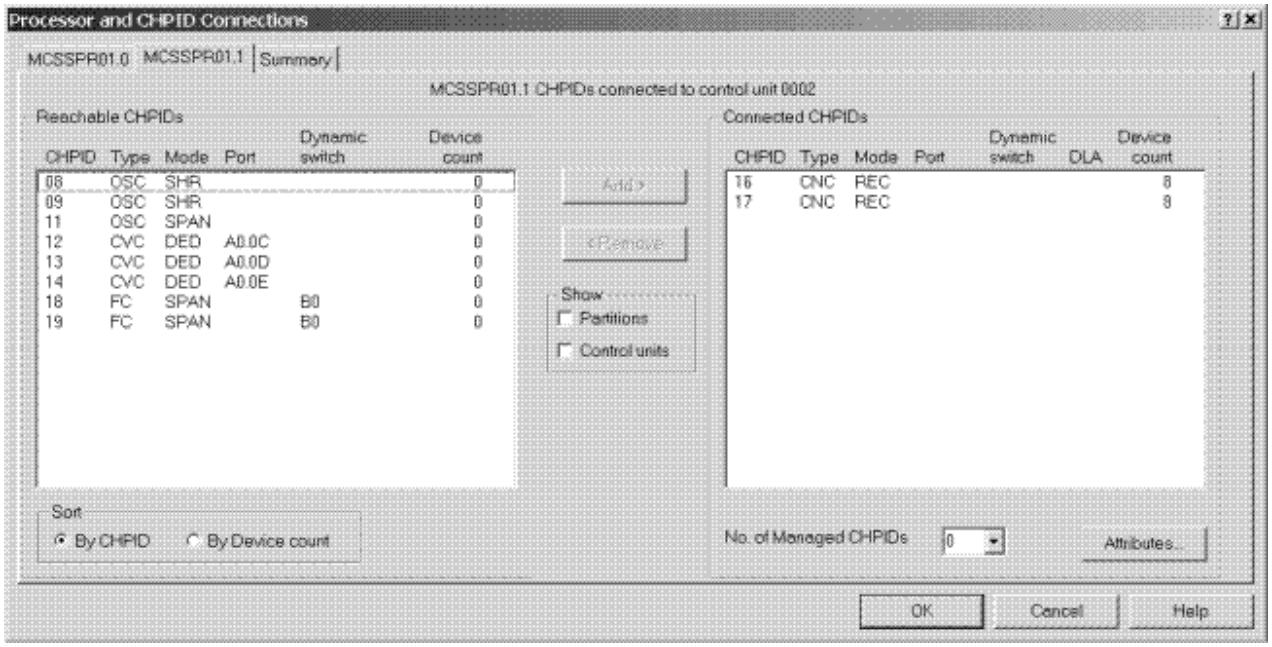


Figure 108. Processor and CHPID Connections

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 176 for more information.

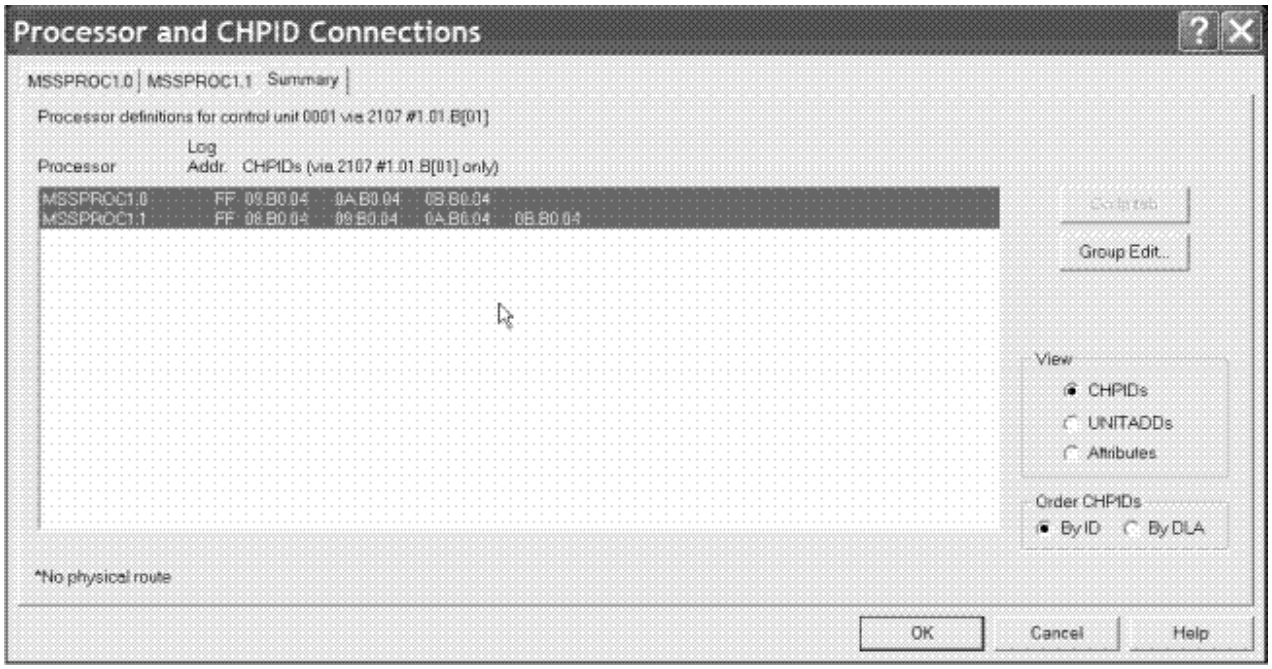


Figure 109. Processor and CHPID Connections

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 want to 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.

If you select a spanned CHPID in the **Reachable CHPIDs** List on the tab, then press the Add button, the **Choose Channel Subsystems** dialog offers other channel subsystem(s) over which the CHPID is spanned, as shown in Figure 110 on page 189. You can then specify the channel subsystem(s) for applying the same operation, so that you do not need to click each tab and press the **Add** button each time.

If you select a spanned CHPID in the **Connected CHPIDs** List on the tab, then press the Remove button, you can also specify the channel subsystems(s) for applying the same operation on the **Choose Channel Subsystems** dialog.

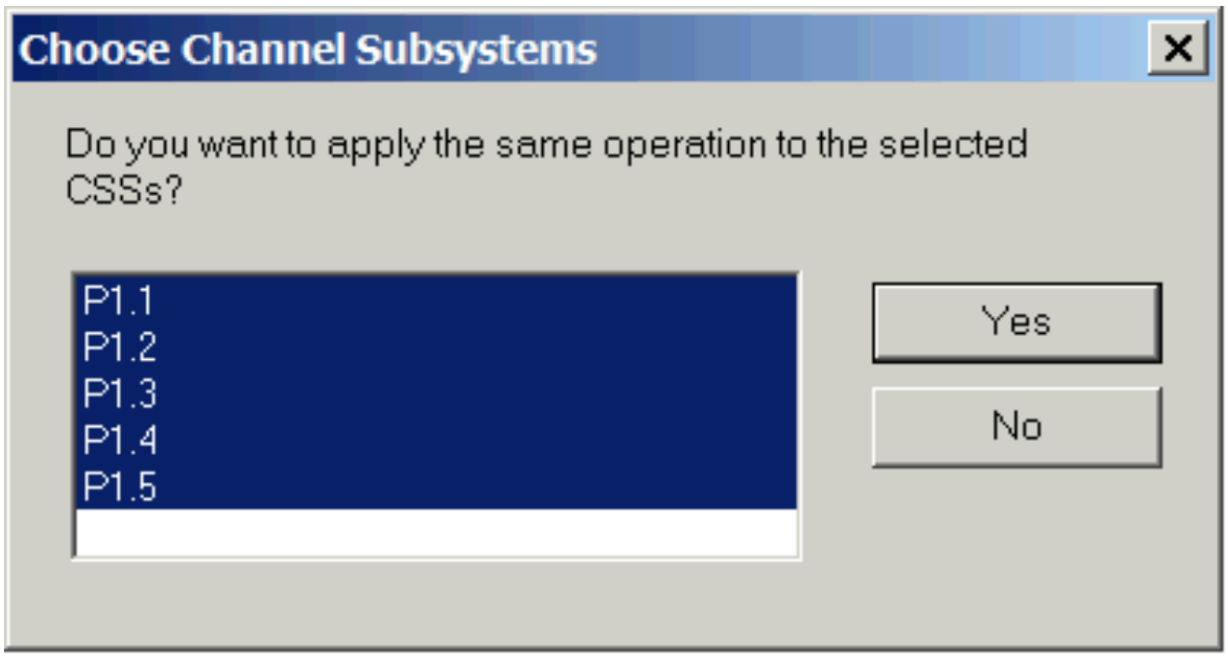


Figure 110. Choose Channel Subsystems

Editing logical definitions between controller and processor

You can edit the attributes for the logical connection between a control unit and a processor in the **Edit Control Unit <=> Processor Definitions** dialog. Press the **OK** button to validate the changes and return to the calling dialog.

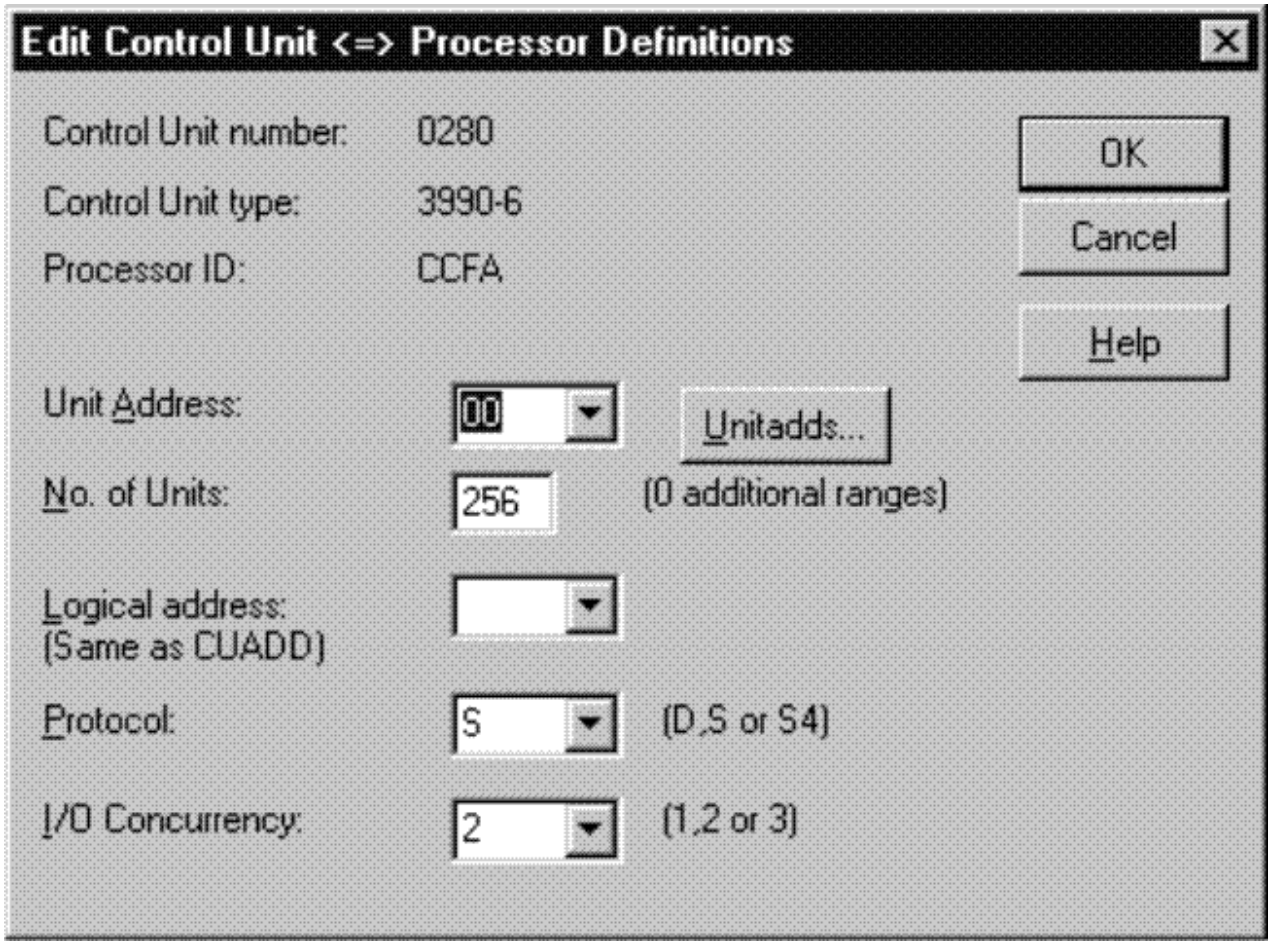


Figure 111. Edit Control Unit <=> Processor Definitions

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 is performed. 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 <=> Processor Definitions** dialog.

Connecting multiple control units to multiple CHPIDs

The ***Add Control Unit <=> Processor Definitions*** dialog allows you to assign CUADD values to the connections of multiple control units in ascending order. You may choose an ***Offset*** (or ***step size***) other than 1 to implement your own numbering schemes.

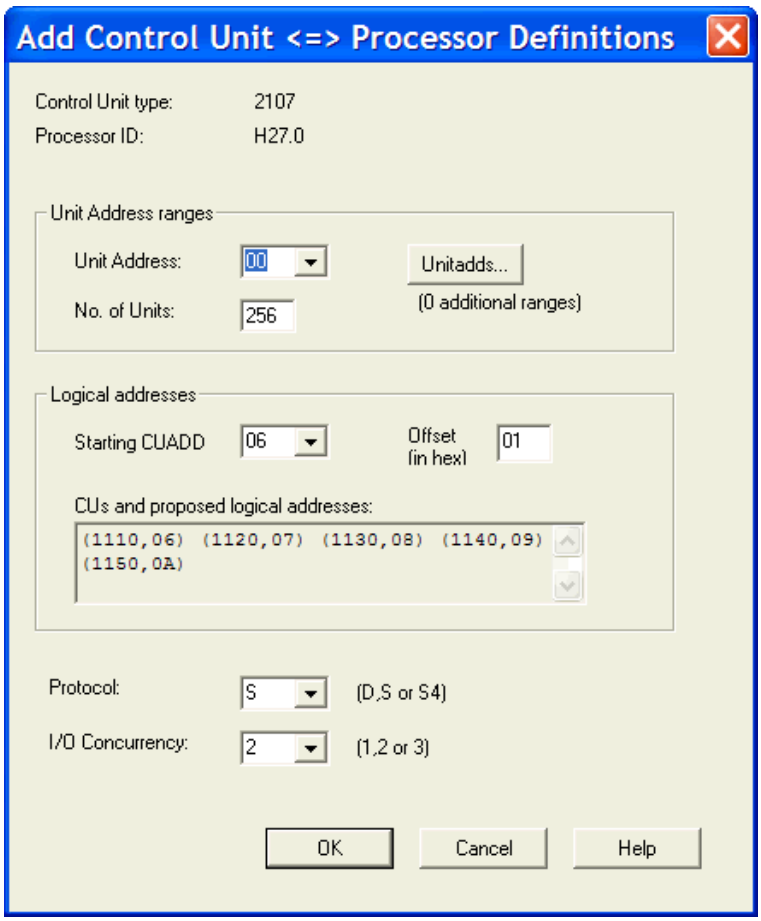


Figure 112. Add Control Unit <=> Processor Definitions

HCM displays this dialog when you establish the first CHPID connection of a control unit. Use this dialog to specify the connection attributes.

For control units that already have processor connections, HCM does not display this dialog. When adding more connections, the previously defined CUADD value is used for the new connection(s).

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 325 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 184). 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 184. 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 193.

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 191. 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 218 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.

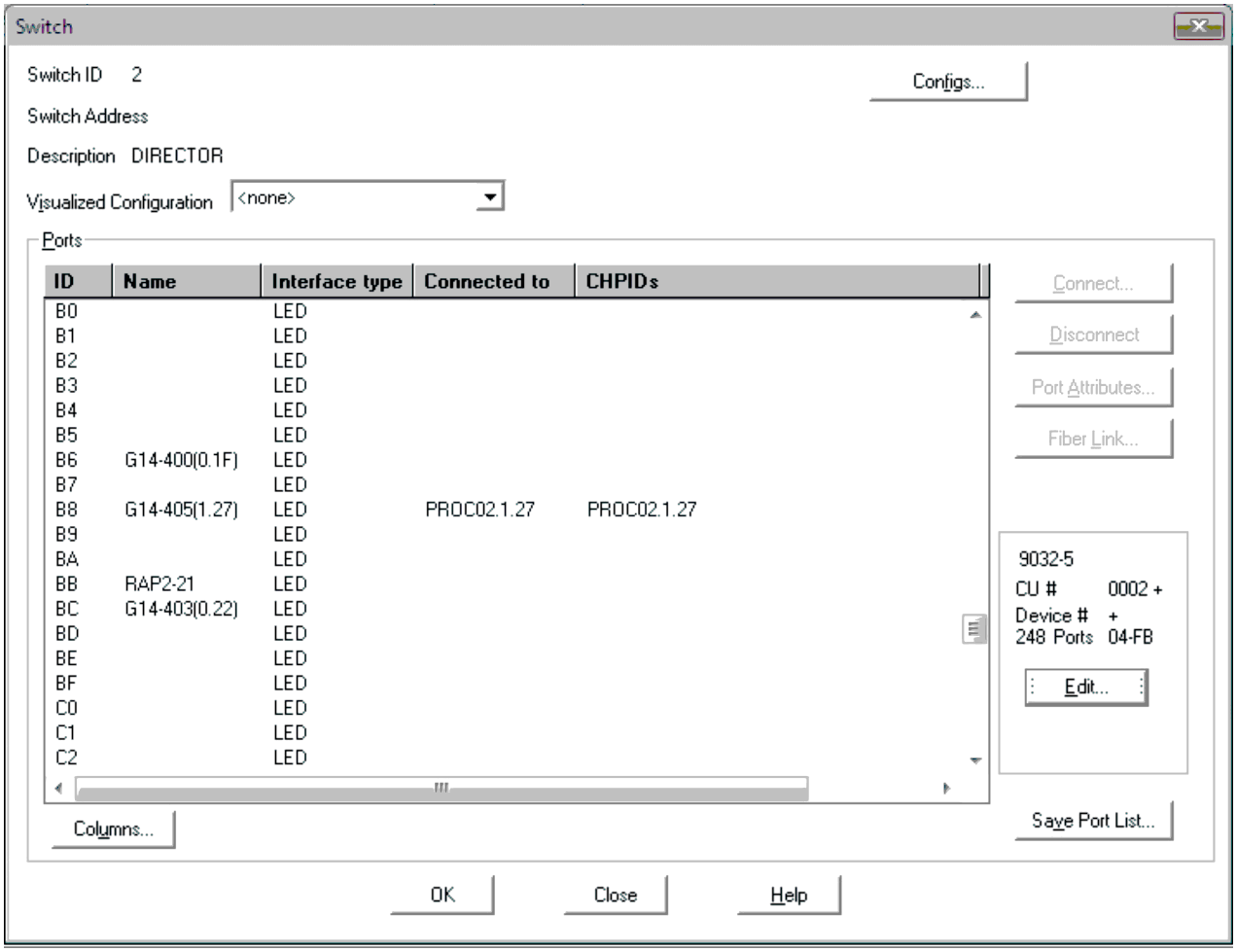


Figure 113. Switch

From the **Switch** dialog shown in Figure 113 on page 194, 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.

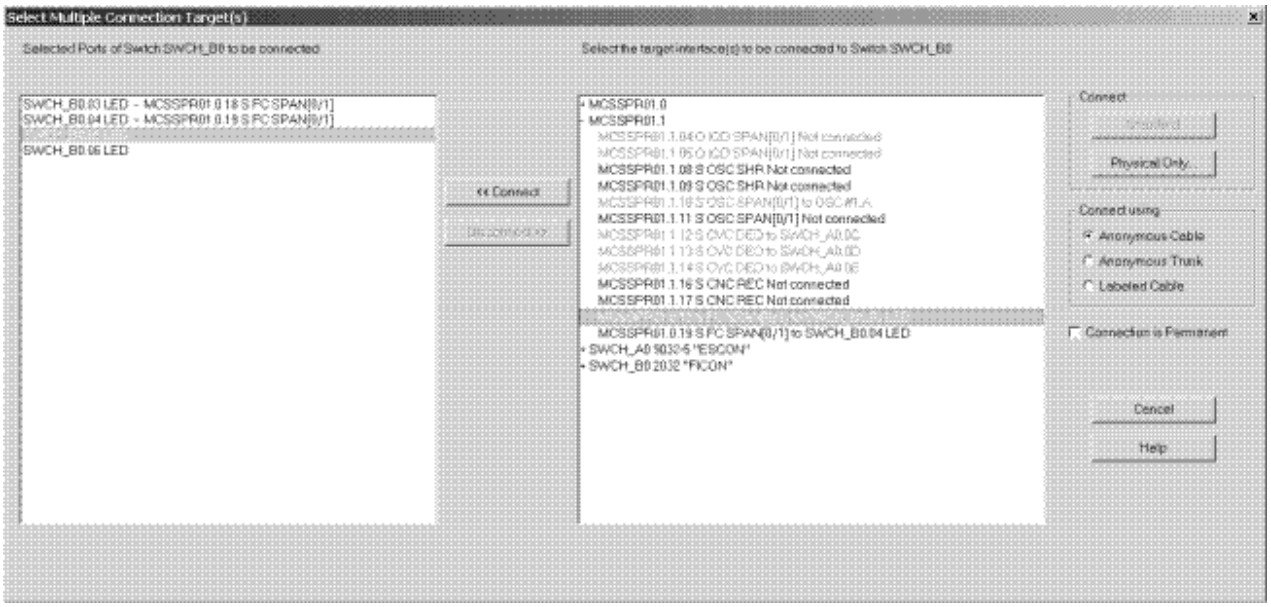


Figure 114. Select Multiple Connection Target(s)

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 115 on page 195) where you can update the dynamic switch for all involved CHPIDs.

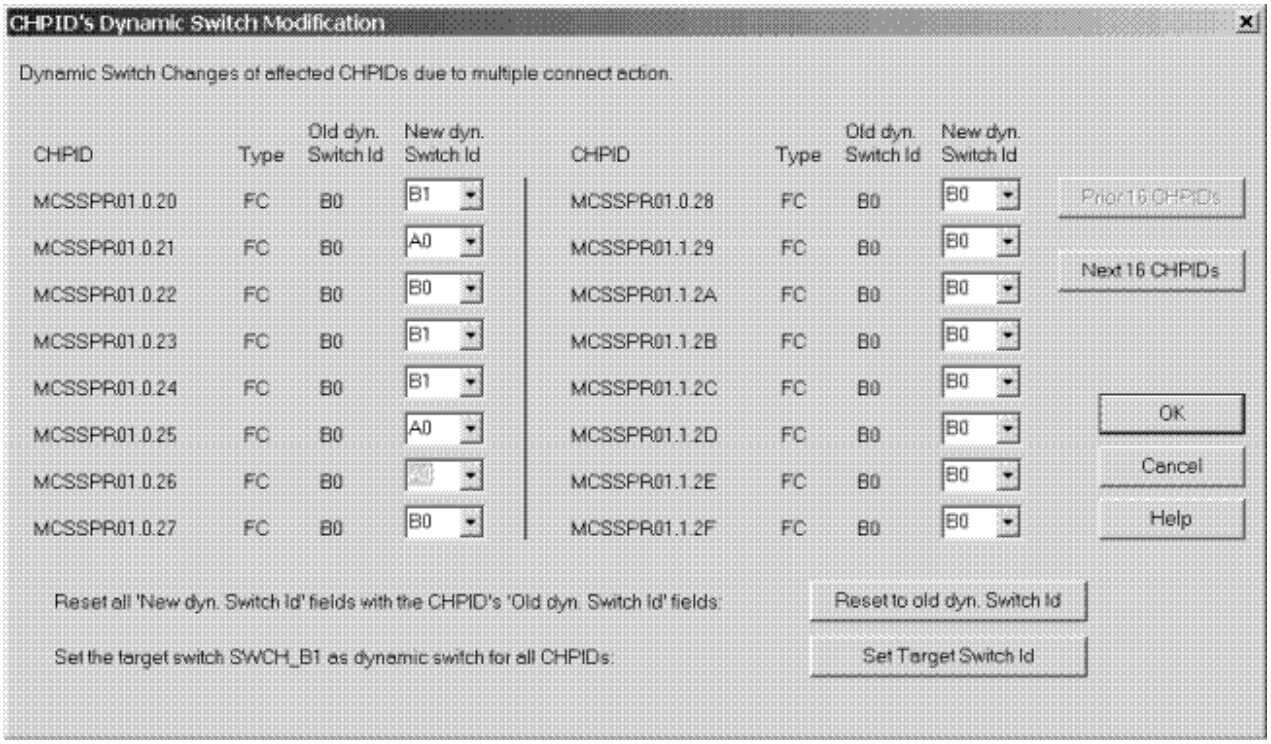


Figure 115. CHPID's Dynamic Switch Modification

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.

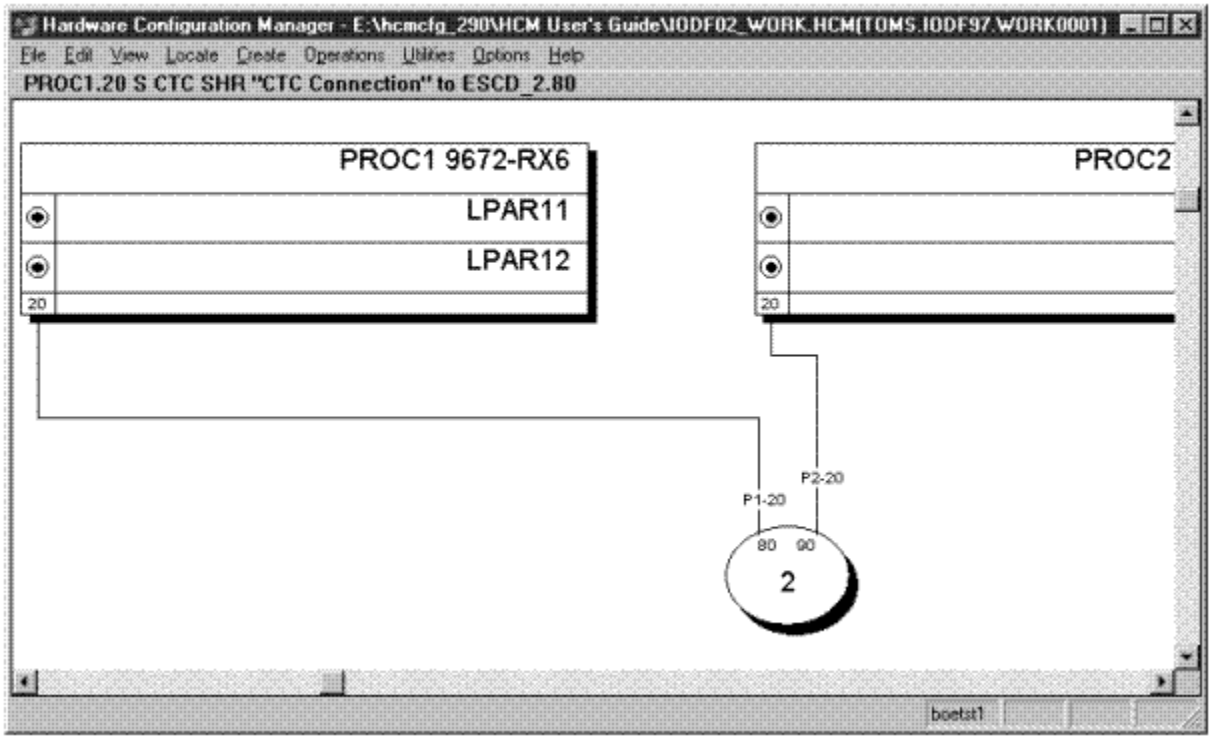


Figure 116. CTC connection between two CHPIDs on different processors through a switch

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.

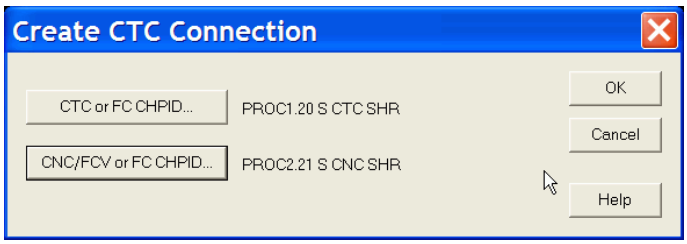


Figure 117. Create CTC Connection

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.

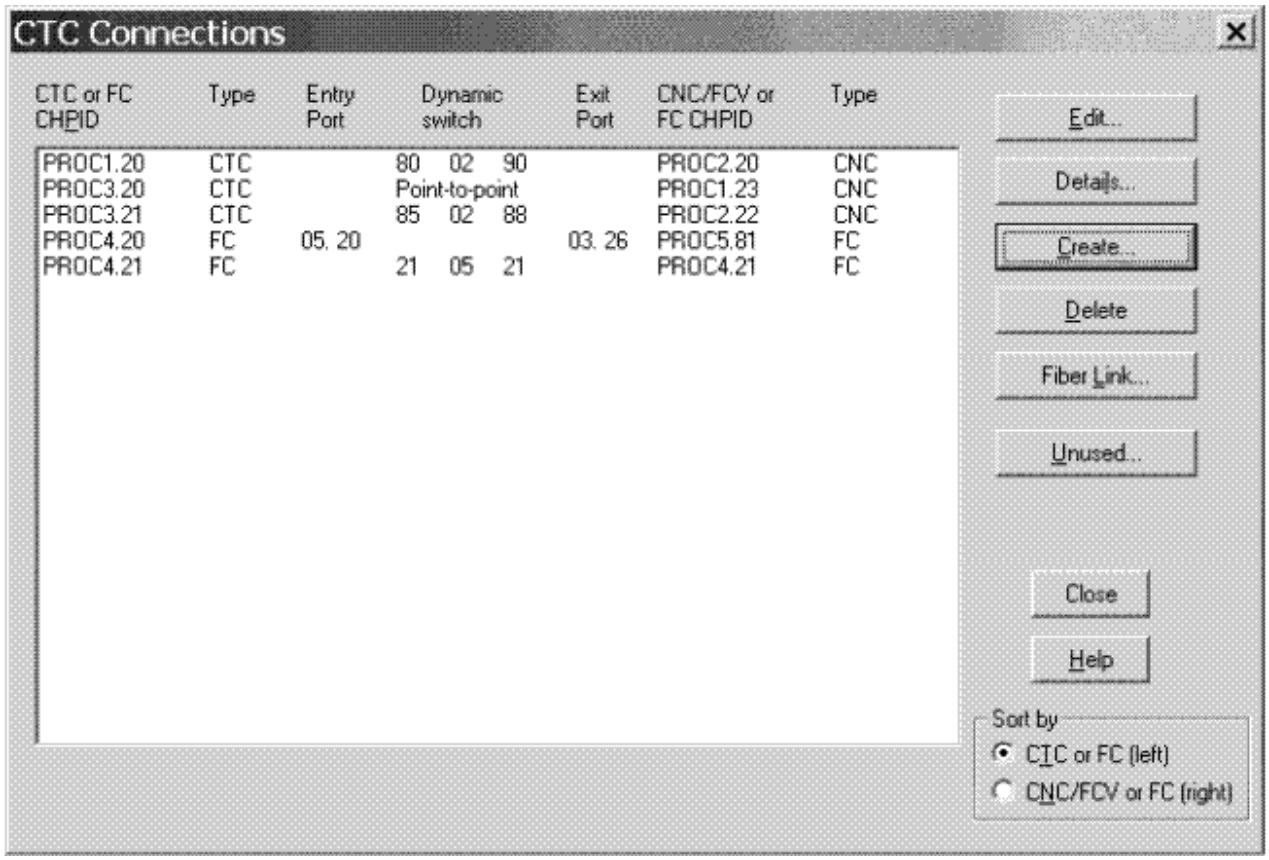


Figure 118. CTC Connections

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.

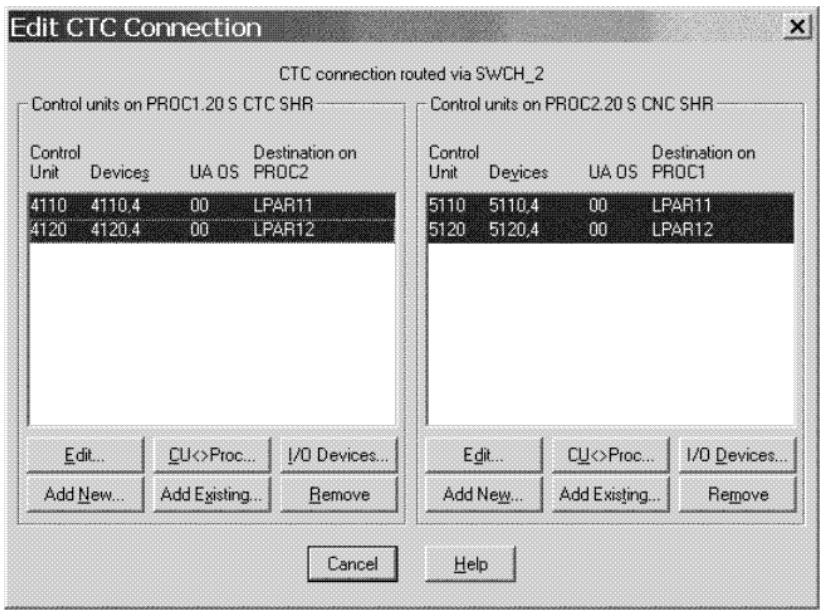


Figure 119. Edit CTC Connection

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 shown:

```
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.

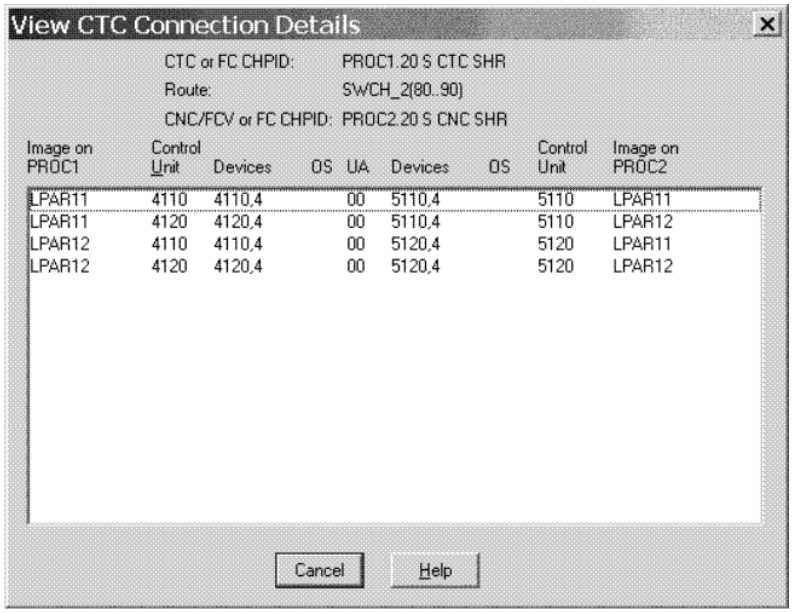


Figure 120. View CTC Connection Details

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 212.

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 appears on the **CTC Connections** dialog. Clicking this button opens the **View Unused CTC Control Units, Devices, and CHPID Connections** dialog.

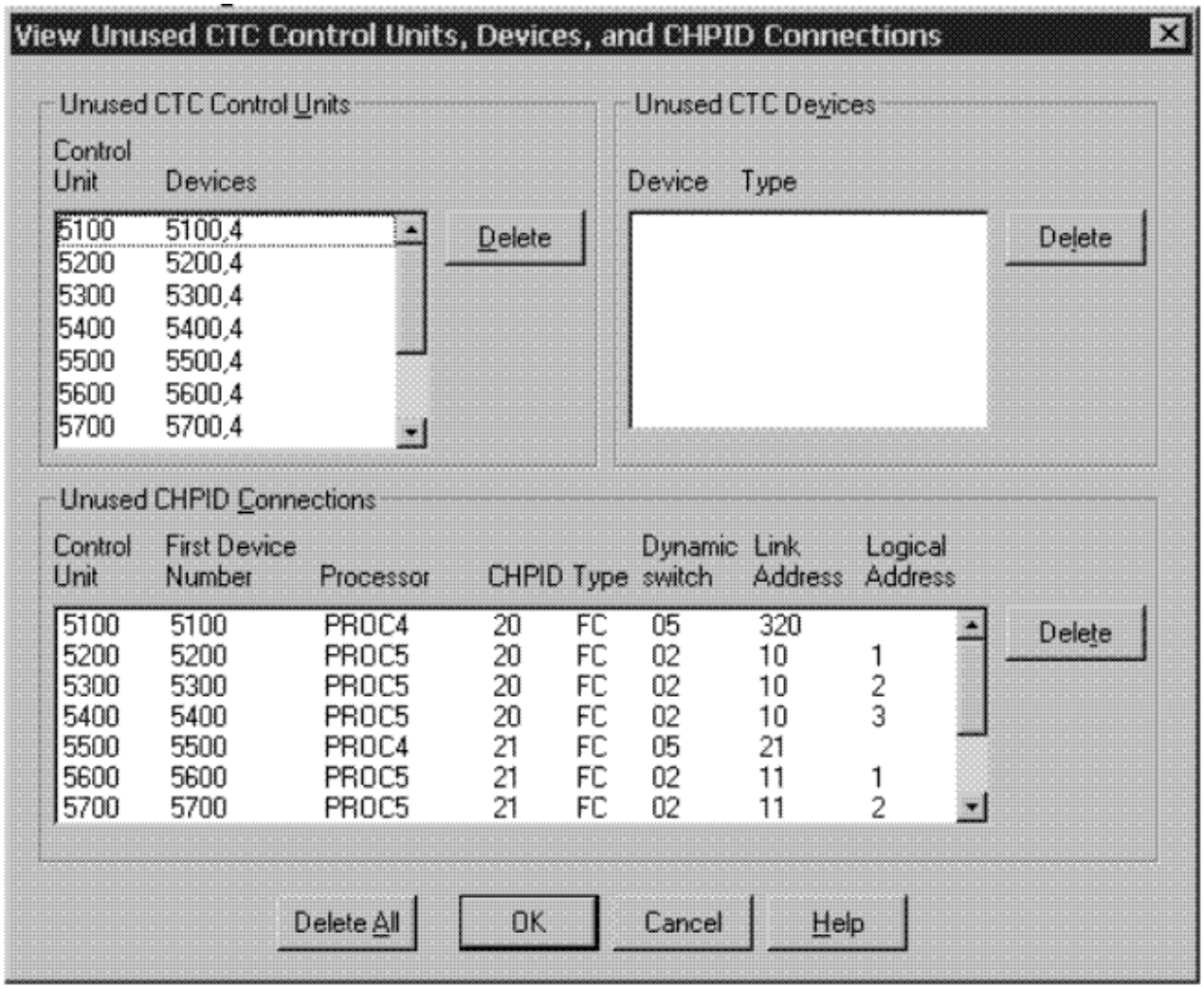


Figure 121. View Unused CTC Control Units, Devices, and CHPID Connections

All invalid (that is not involved in a valid CTC connection) CTC-related IODF objects 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, adhere to the following precautions:

- 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 57.
- 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 text reports and diagram reports” on page 266 (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 120 on page 199. 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 200.

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.

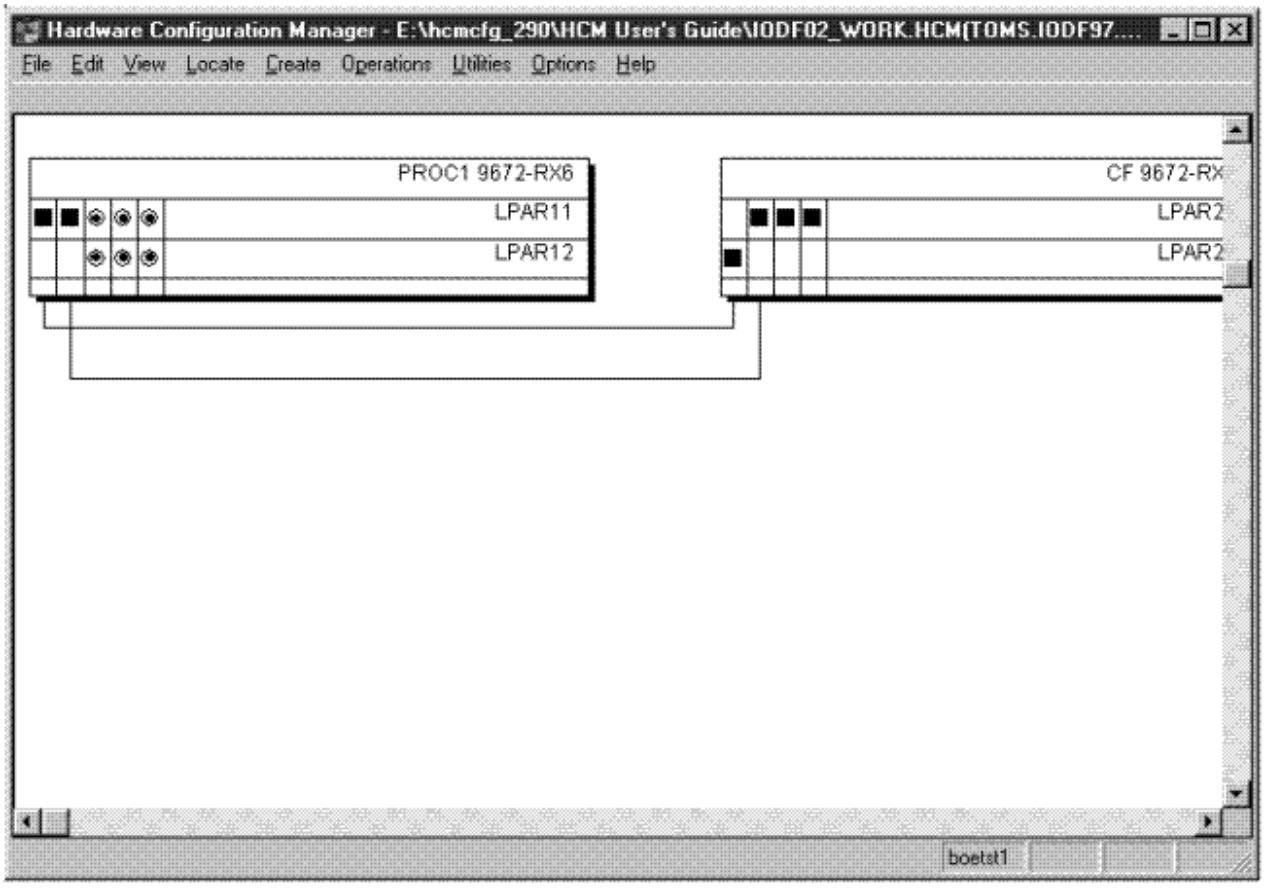


Figure 122. Coupling Facility Connection

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.

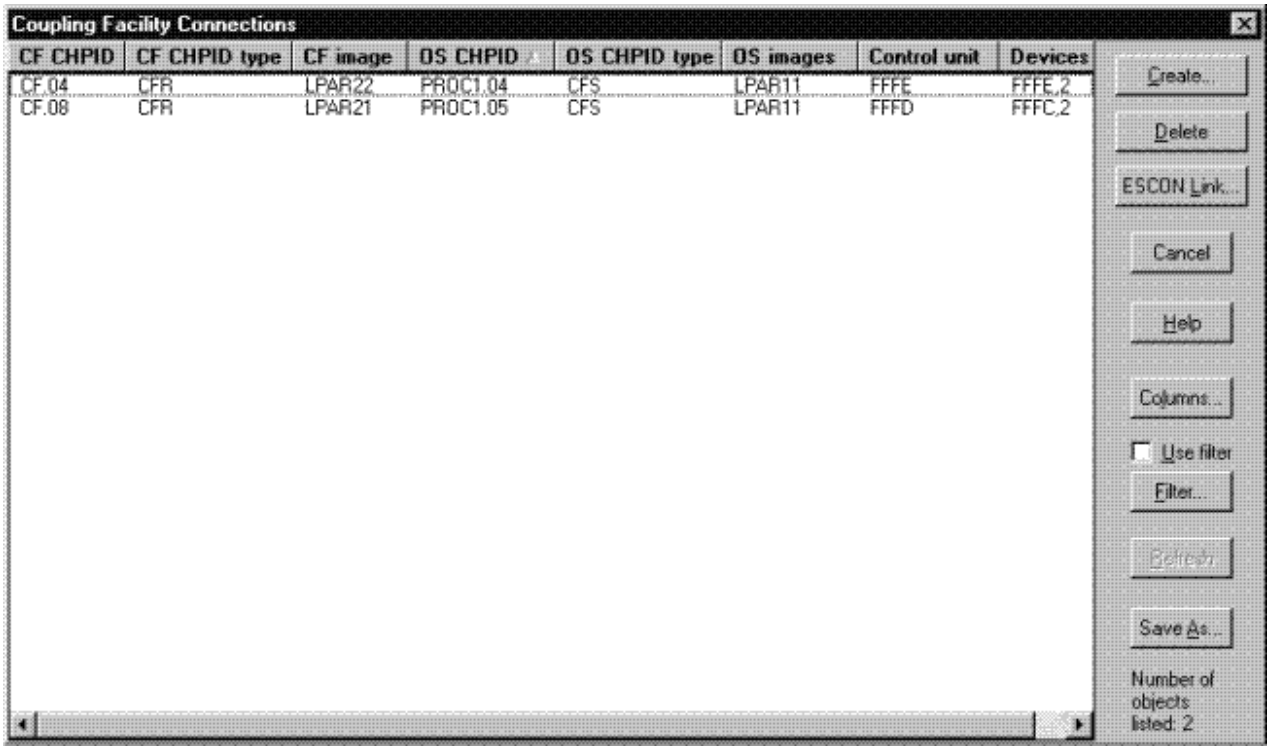


Figure 123. Coupling Facility Connections

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.

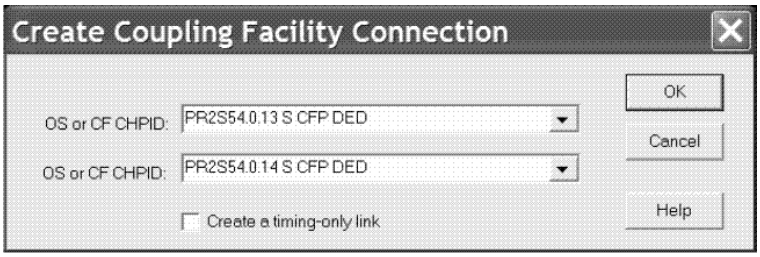


Figure 124. Create Coupling Facility Connection

With this dialog, HCM also supports Server Time Protocol (STP) links (timing-only links) between two zSeries (z890, z990 or later) processors. Timing-only links are only needed in case coupling links are not desired, not possible or not needed. If you want to define a coupling facility connection which will be used as a timing-only link, select the ***Create a timing-only link*** checkbox. Both processors must be timing capable in this case and the used channel paths must either be CFP or CBP.

For an STP only link, HCD generates a control unit of type STP on both sides of the connection. No devices are defined.

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.

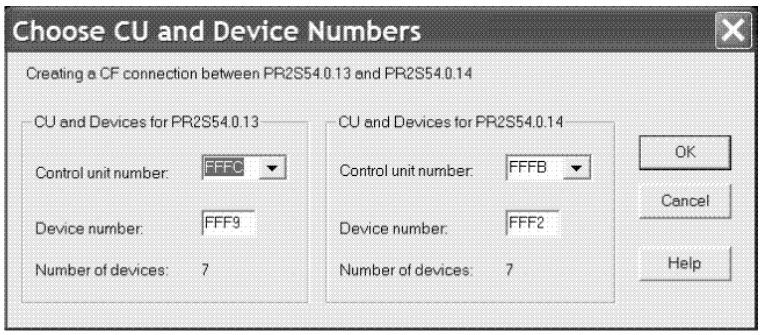


Figure 125. Choose CU and Device Numbers

For a timing-only link, the dialog from Figure 125 on page 206 would contain a text *Timing-only Link* instead of the *Device number* field and set the number of devices to zero.

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 175). 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 reused in the near future.
- The inventory can be populated by importing cable descriptions from an ASCII file. See Chapter 14, “Importing and exporting data,” on page 333 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.

© Copyright IBM Corp. 1995, 2021

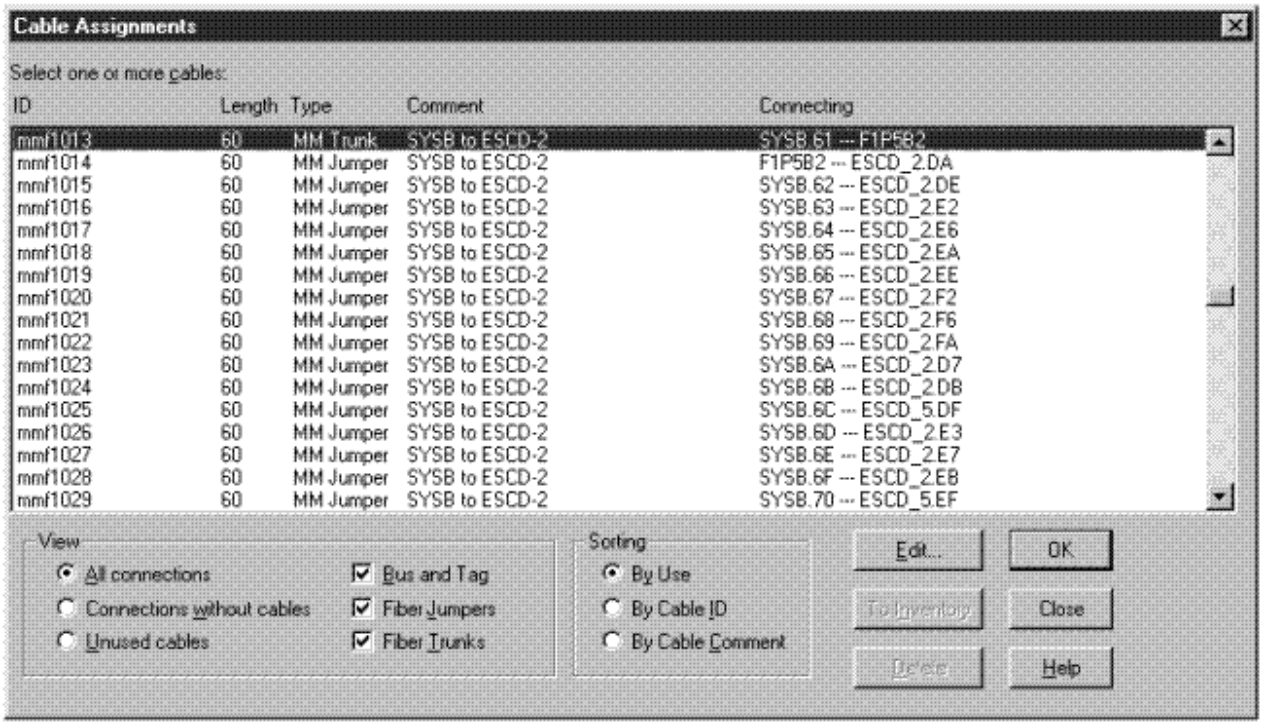


Figure 126. Cable Assignments Dialog

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. Unused cables include 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 does not retain any information about these cables, once they are deleted.

Note: You can only delete a cable if it is unused. If you want to delete a cable which is in use, you first have to unassign the cable from the current fiber link.

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 212 for more details.

Creating and editing cables

Use any of the methods described under “Assigning cables to connections” on page 207 to open the **Assign Cable** dialog for a particular connection.

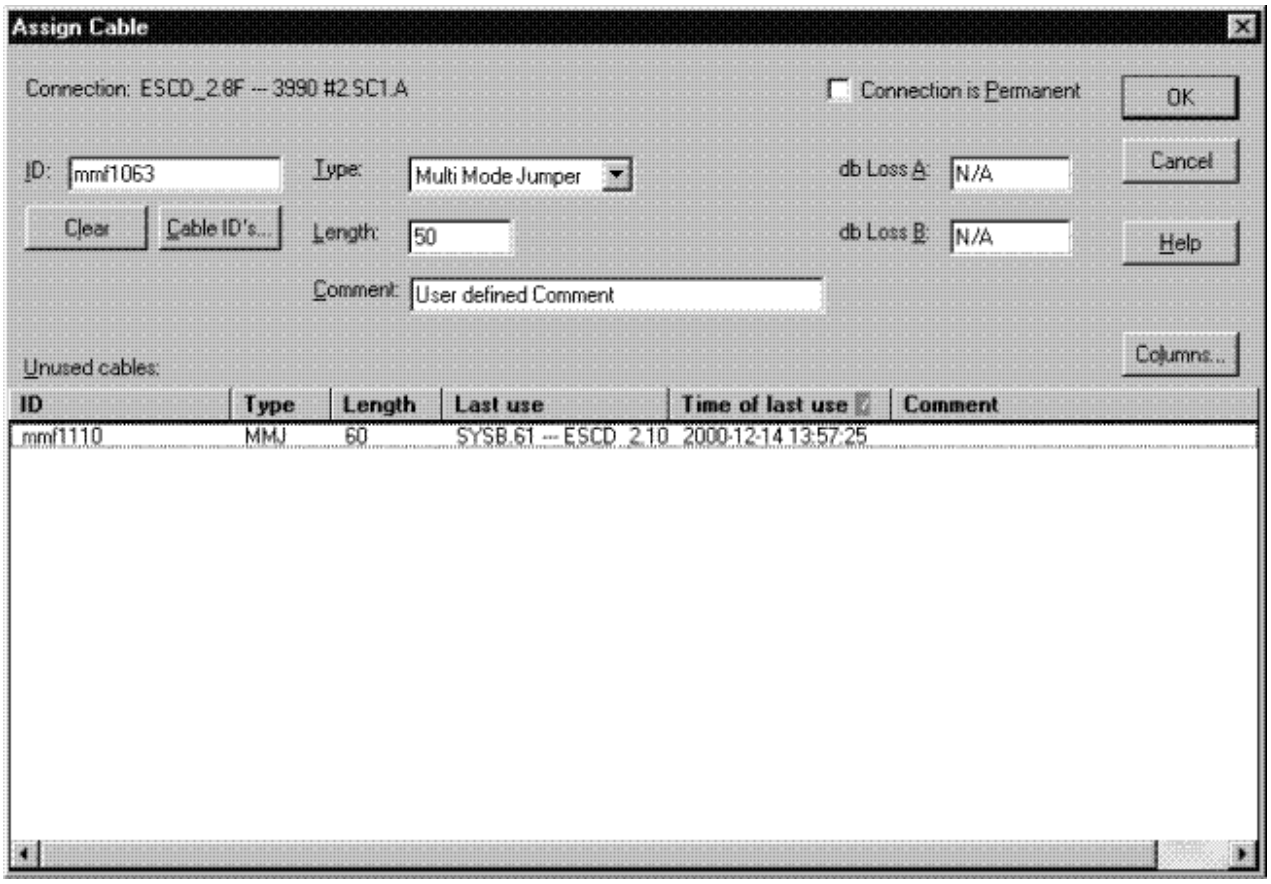


Figure 127. Assign Cable Dialog

Creating a new cable to assign to a connection

Perform this task using the **Assign Cable** dialog which can be reached as follows:

- **Edit --> Cable Assignments...** opens the **Cable Assignments** dialog, where you select a connection without a cable and press **Edit**.
- From a connection's context menu, you select **Edit Cable...**.

Then, 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 press **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
Bidirectional

Serial

Single Mode Jumper
Multi Mode Jumper
Single Mode Trunk
Multi Mode Trunk

Note: HCM supports different interface types for CHPIDs and controller interfaces and also supports different cable types for serial and parallel interface types (see the previous list). If, for example, you want to assign a cable to an OSA CHPID type, you first have to change the interface type from OSA to seriell or parallel. For information about how to do this, refer to "CHPID interface type" on page 119 and "Changing the channel interface type" on page 137.

This involves, for example, that you can assign a cable definition also for an internal CHPID like IQD or ICP, where no physical cable can be attached to. Such cable definitions in HCM allow you to further document your physical environment.

Reusing an existing cable for a connection

The Unused Cables listbox in the **Assign Cable** dialog 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 and then press **OK**. (Alternatively, type its cable ID in the appropriate text box, and press **OK**).

For verification of unique cable IDs, press the **Cable ID's...** button to display 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 this connection in the **Cable Assignments** dialog and press the **Edit** button. This opens the **Assign Cable** dialog where you press the **Clear** button and then the **OK** button. Back in the **Cable Assignments** dialog, you can now find the cable in the 'Unused cables' view with no cable assignment.

You can also reach the **Assign Cable** dialog from a connection's context menu by selecting **Edit Cable...**.

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.

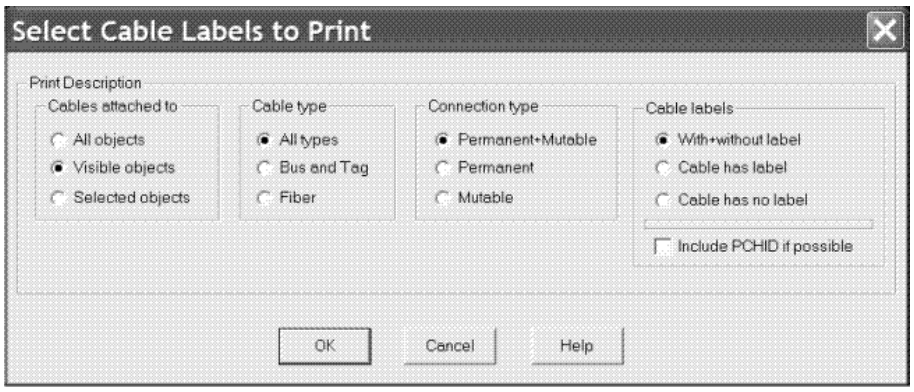


Figure 128. Select Cable Labels to Print Dialog

Select the cables for which you want to print labels. Choose a combination of three cable attributes: cable type, connection type, and cable labels. Using the radio buttons in the first box, you can restrict the number of labels by printing only those attached to visible objects (those currently included in the diagram), or only those attached to currently selected objects.

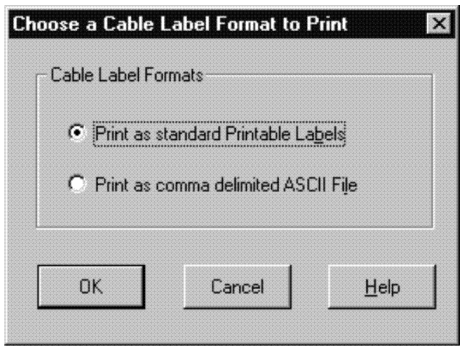


Figure 129. Choose a Cable Label Format to Print Dialog

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.

If you select the ***Include PCHID if possible*** check box from the dialog shown in Figure 128 on page 211, then, if applicable, the PCHID of the channel that is connected by the cable, is also included in the output (in parenthesis), like for example in a comma delimited ASCII file:

without PCHID

```
ERV01.3.98,2105 #1.6.A[24],2006-05-31,c_1234,ERV01.3.98,2105 #1.6.A[24]
2105 #1.6.A[24],ERV01.3.98,2006-05-31,c_1234,ERV01.3.98,2105 #1.6.A[24]
```

with PCHID

```
ERV01.3.98(398),2105 #1.6.A[24],2006-05-31,c_1234,ERV01.3.98(398),2105 #1.6.A[24]
2105 #1.6.A[24],ERV01.3.98(398),2006-05-31,c_1234,ERV01.3.98(398),2105 #1.6.A[24]
```

See “Sample cable labels” on page 276 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 175, 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 a 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 about disconnecting fiber links. For more information about step 3, see Chapter 7, “Connecting objects in your configuration,” on page 175.

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

If a patchport is connected to a CHPID or switch port in this way, 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. Furthermore, 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:

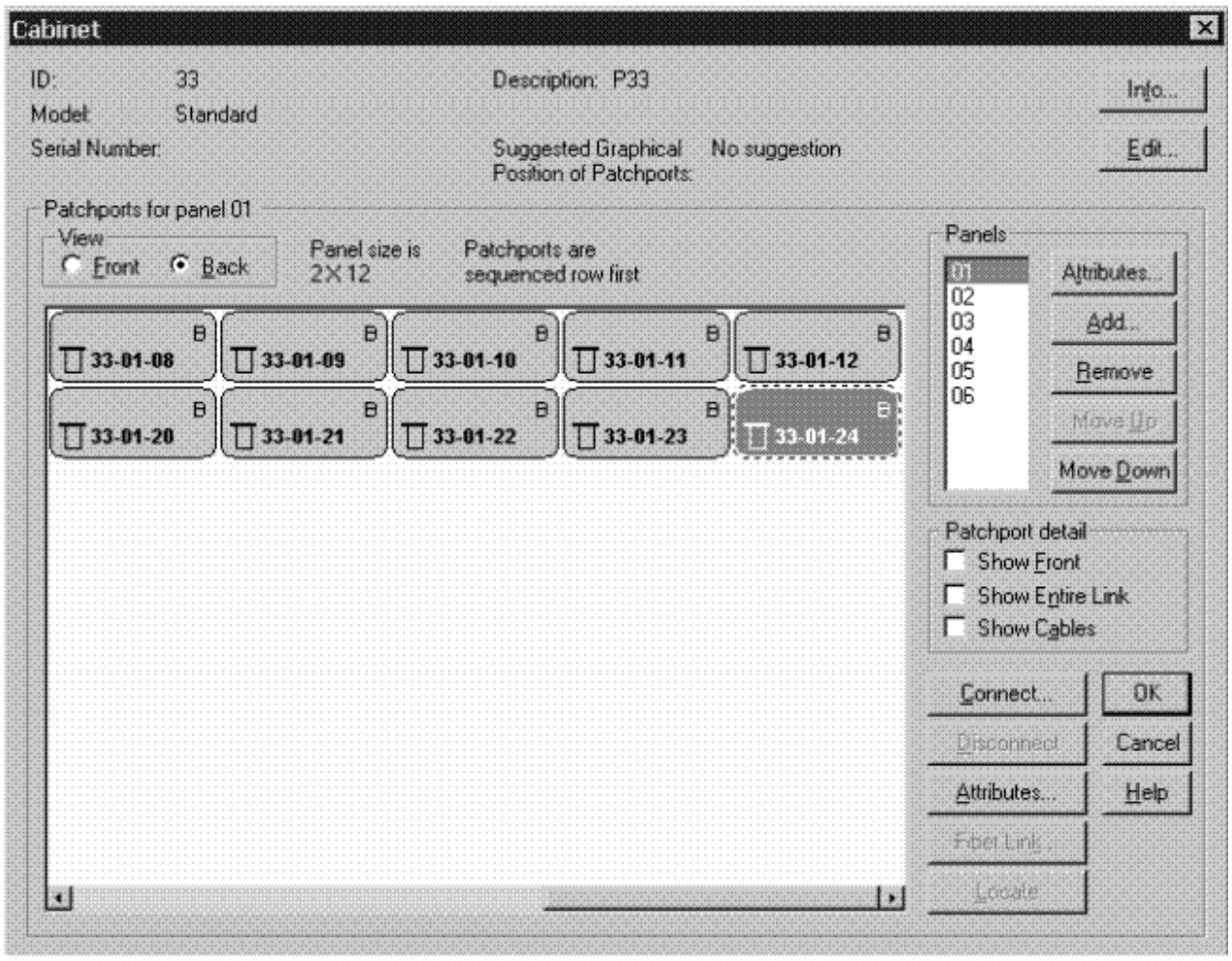


Figure 130. Cabinet Dialog

Make sure the **Back** radio button in the **View** group box is selected. This tells 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. Now click **Connect...**. HCM opens the **Select Connection Target(s)** dialog.

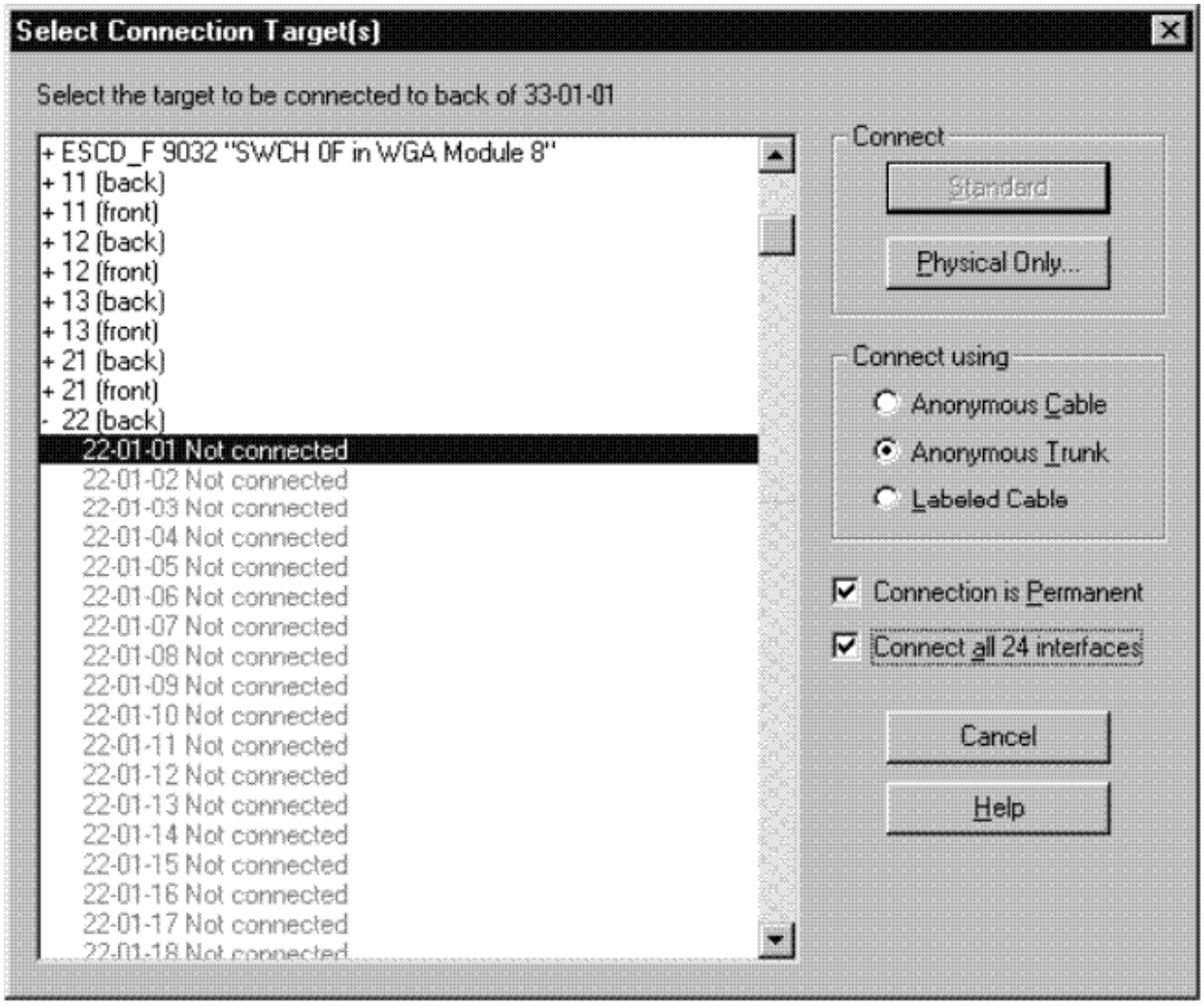


Figure 131. Select Connection Target(s)

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 132 on page 215. 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 132 on page 215 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.

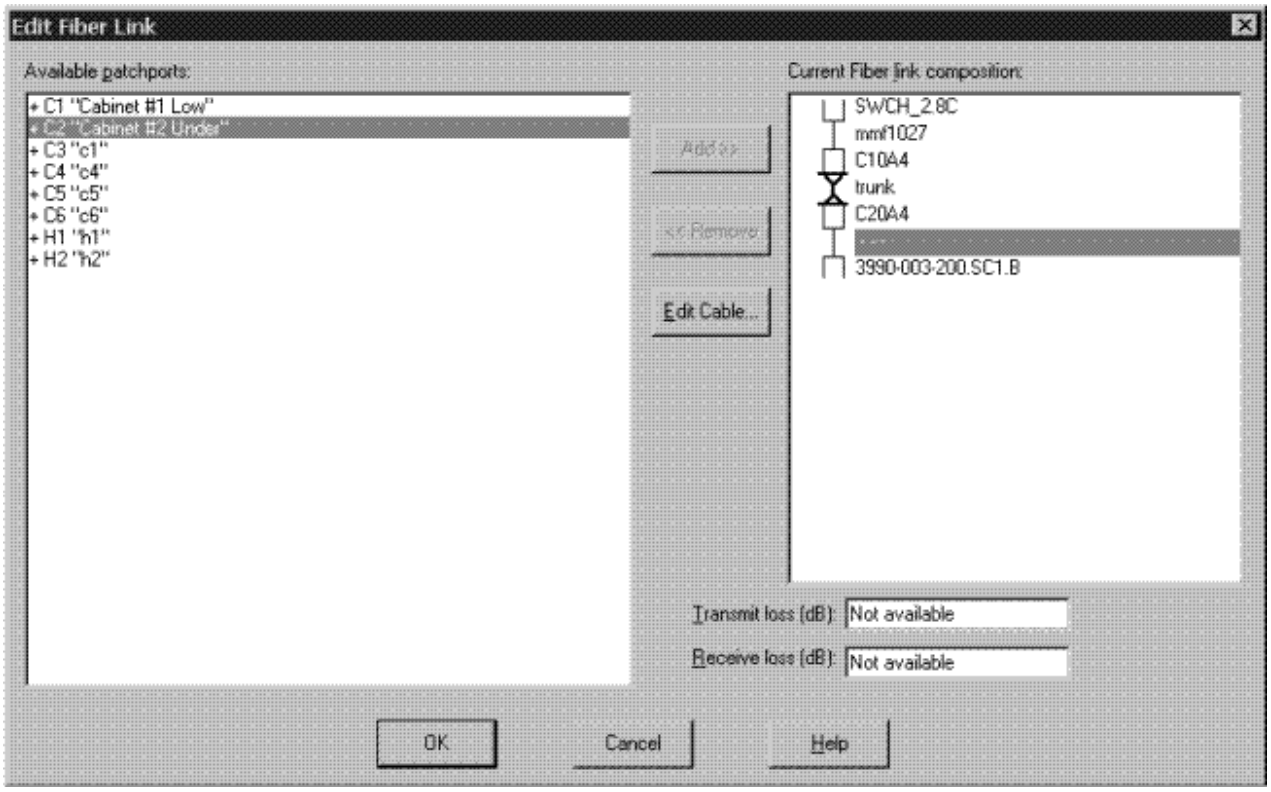


Figure 132. Edit Fiber Link

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 207 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 the Configuration Diagram 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:

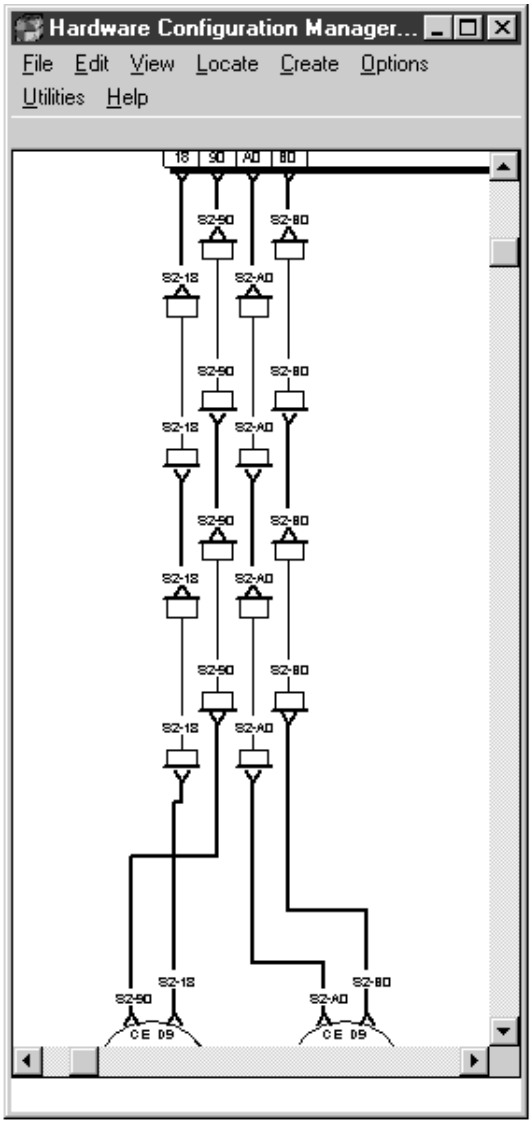


Figure 133. Configuration Diagram

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:

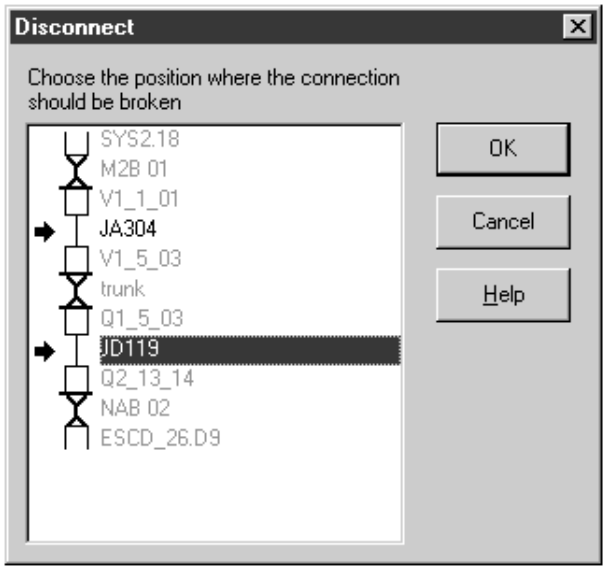


Figure 134. Disconnect Dialog

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 (for example, there is only one jumper connecting the CHPID and port). In such cases, the connection is 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 183 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 176 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 for performing time-consuming configuration tasks, which you can invoke from the **Utilities** menu. These utilities support you in making complex configuration changes without violating any logical definitions that are valid in your configuration. They are wizards that guide you through the various functions. At the end, you see a summary of your changes where you can accept or discard them. You can also save the summary to a file. These utilities are described in the following sections of this chapter:

- "Performing I/O Autoconfiguration" on page 221
- "Moving port connections" on page 231
- "Aggregating CHPIDs" on page 233
- "Converting CNC channels to FICON channels" on page 233
- "Changing the CHPID link address format" on page 234
- "Creating/Copying/Editing an I/O subsystem" on page 234

Besides these utilities that assist you in configuration tasks, you can use further utilities for special purposes:

- To assign physical channel identifiers (PCHIDs) to logical CHPIDs, use the **CHPID Mapping Tool Support** (described in "Using the CHPID Mapping Tool Support utility" on page 73).
- To assign world-wide port names to the FCP device paths, use the **WWPN Prediction Tool Support** (described in "Using the WWPN Prediction Tool Support" on page 242).
- To create an IOCP input file or an I/O configurations file by extracting data from the currently accessed IODF for later migration into a different target IODF, you can use the ***Export IOCP or I/O Configuration statements*** utility (described in "Exporting IOCP or I/O configuration statements" on page 235)
- To check configuration files for structural consistency, use the **Configuration file checker** (described in "Checking a configuration" on page 76).
- To create an exported MCF and send it to IBM Support for MCF problem diagnosis, select **Exporting Master Configuration File** (described in "Exporting an MCF" on page 390).
- To examine performance data of objects in your active sysplex, and to discover potential bottlenecks, select menu item **Browse Performance Data...** (described in "Browsing performance data" on page 237).
- To migrate I/O configuration statements from an existing input file into a selected work IODF, use the **Import I/O Configuration Statements** dialog (described in "Importing I/O configuration statements" on page 236).

You invoke all available utilities described in this chapter from the **Utilities** menu, except for the following ones:

- Invoke the **Create I/O Subsystem** utility from the **I/O Subsystem...** pull-down in the **Create** menu bar choice.
- Invoke the **Copy I/O Subsystem** and **Edit I/O Subsystem** utilities from a controller's context menu.
- Invoke the **Import I/O Configuration Statements** dialog by pressing the ***Import...*** button in the ***Cfg. Statements*** section from the **IODFs** dialog (Figure 24 on page 60).

Performing I/O Autoconfiguration

The ***Autoconfiguration*** wizard discovers and automatically defines switched FICON connected DASD and tape control units and devices. The definition of devices and control units is guided by user-defined policies. You can either accept the proposed definitions without changes, or you can update the changes before committing them to your IODF. The wizard is available only when HCM is connected to a z/OS system and a work configuration is opened.

© Copyright IBM Corp. 1995, 2021

I/O Autoconfiguration tasks are similar between HCD and HCM. See How to work with I/O Autoconfiguration in the *z/OS HCD User's Guide* for more information.

Before using I/O Autoconfiguration in HCM ...

... it is recommended to read topic "Preparing the IODF" on page 57. In particular, you should specify correct and matching serial numbers for control units residing in controllers that are chosen for autoconfiguration. This means that the control unit serial numbers should match the number of the containing controller.

When loading an IODF, HCM groups control units into controllers guided by their serial numbers. Established groupings are normally not changed by IODF resynchronization. If the grouping in the HCM configuration does not match the actual physical grouping of the control units, I/O Autoconfiguration may be unable to correct this. In this case, newly established logical connections will often result in footnotes for the affected control units: ***control unit xxxx has logical connection(s) without a physical route.***

How to start I/O Autoconfiguration

To start the wizard, select ***Utilities ---> Autoconfiguration...*** from the menu bar. This invokes the wizard's ***Welcome*** page. The **Next** button proceeds to the ***Autoconfiguration - Options*** page (Figure 135 on page 223).

You may want to check the configuration policies in effect by pressing the ***Policies...*** button on this page. Details on how to define these policies are in topic "How to define autoconfiguration policies" on page 228.

How to define the scope and mode of discovery

On the ***Autoconfiguration - Options*** page (Figure 135 on page 223), select the scope of discovery to determine, which controllers should be detected. In the ***Scope of discovery*** section, you can select from:

New controllers only
: This indicates that HCM should discover new controllers only.

All controllers
: This indicates that HCM should discover new and changed controllers.

Controller containing CU:
: This indicates that HCM should discover the controller containing the specified control unit only. This specified control unit must be a DASD or tape control unit that is already defined in the IODF.

Controller with serial number:
: This indicates that HCM should only discover a single controller matching the specified serial number or multiple controllers matching a serial number pattern. You can specify a complete serial number, or look for multiple controllers using a wildcard (*) as prefix or suffix or both of the pattern. For example, F30WD, F3*, *F3*, *WD, or *WD* are valid values to include controller F30WD.

Control units in a controller are discovered by sequentially checking for configured CUADDs. If the configured CUADDs have gaps, you should use the ***Full mode discovery*** option. Otherwise, discovery stops after several consecutive unused CUADD values that do not exist on the target controller.

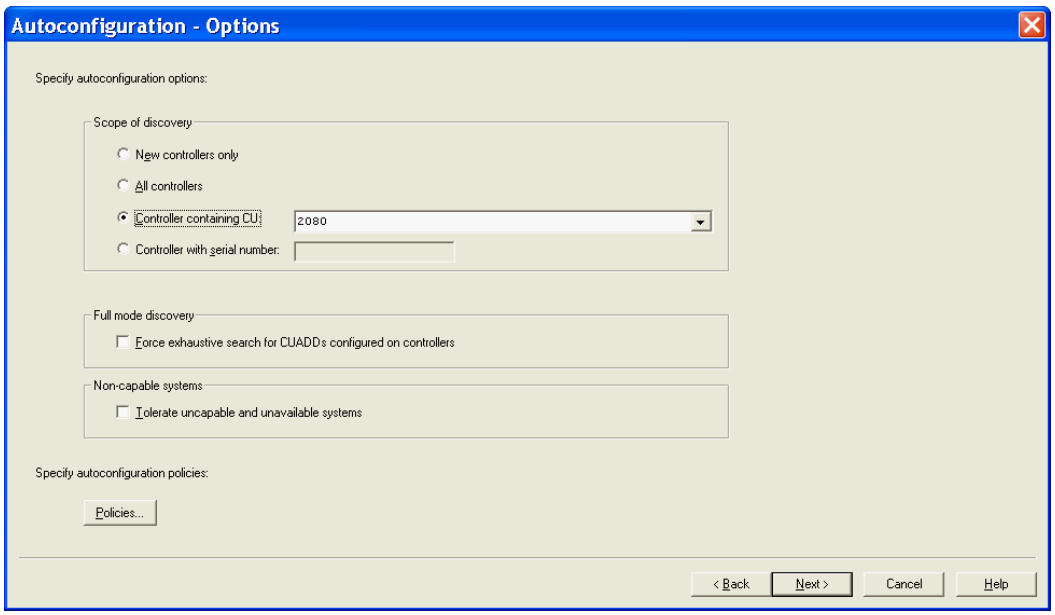


Figure 135. Autoconfiguration - Options

With the ***Policies...*** button you can invoke a dialog to set or edit user-defined autoconfiguration policies as described in "How to define autoconfiguration policies" on page 228.

How to work with discovered controllers

With appropriate configuration policies and options defined, press the ***Next*** button on the ***Autoconfiguration - Options*** page (Figure 135 on page 223). The wizard now shows the ***Discovered Controller List*** (Figure 136 on page 223). It acts as a focal point for I/O Autoconfiguration: here you select an operation mode and controllers to be configured. And you will return to this page, either to select more controllers for configuration or to exit the wizard when you are done.

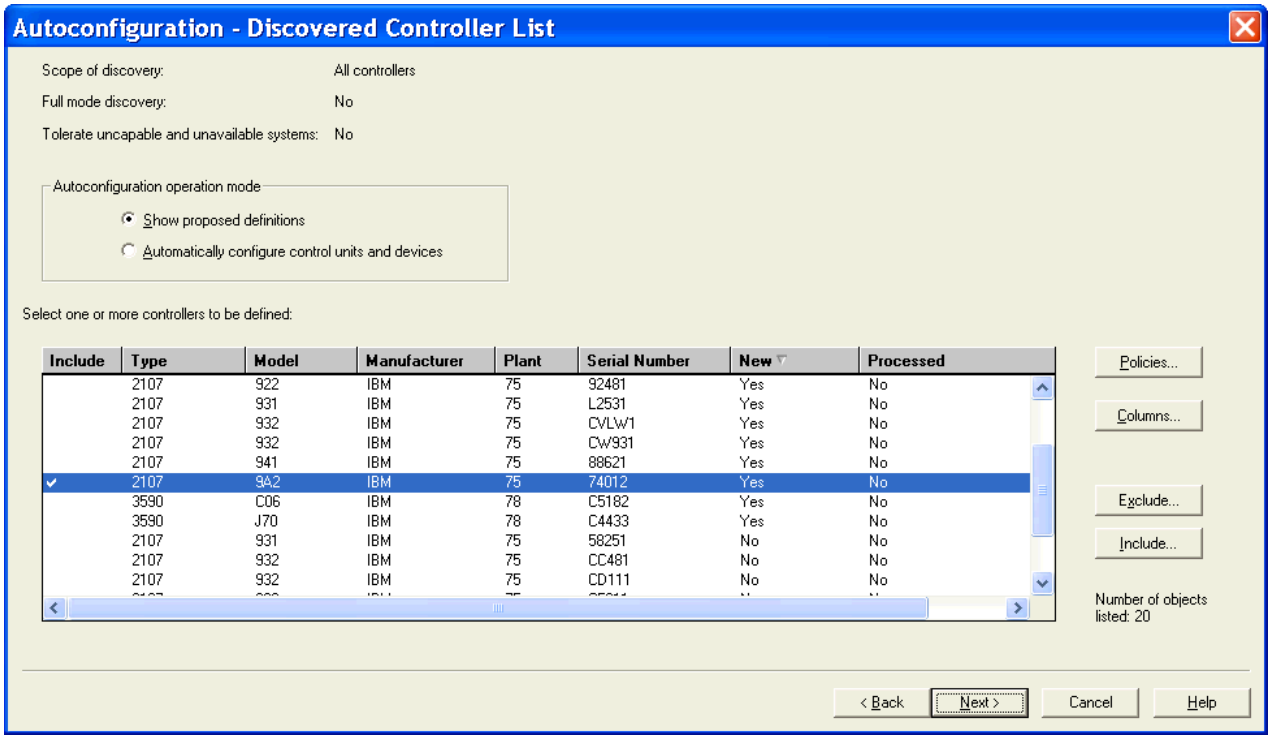


Figure 136. Discovered Controller List

Now choose the ***Autoconfiguration operation mode***:

- ***Show proposed definitions*** selects the **attended mode**:

  Autoconfiguration is done for one selected controller at a time. HCM presents all proposed definitions and you can change them. You can accept or reject all definitions made for the controller on the ***Summary*** page (Figure 143 on page 228).

- ***Automatically configure control units and devices*** selects the **unattended mode**:

  Autoconfiguration makes definitions for all selected controllers quietly. The definitions are presented on the ***Summary*** page, but cannot be rejected.

Select one or more controllers from the list and press the ***Include...*** button to mark them for autoconfiguration. The column ***Include*** shows a check mark for all selected controllers. The ***Next*** button is enabled when at least one controller is marked. Pressing this button starts autoconfiguration for each marked controller in turn.

The wizard now proceeds depending on the chosen operation mode.

How to work with proposed control units

The wizard page ***Proposed Control Unit List*** is shown in attended mode only. It presents the control units for the one controller currently being configured.

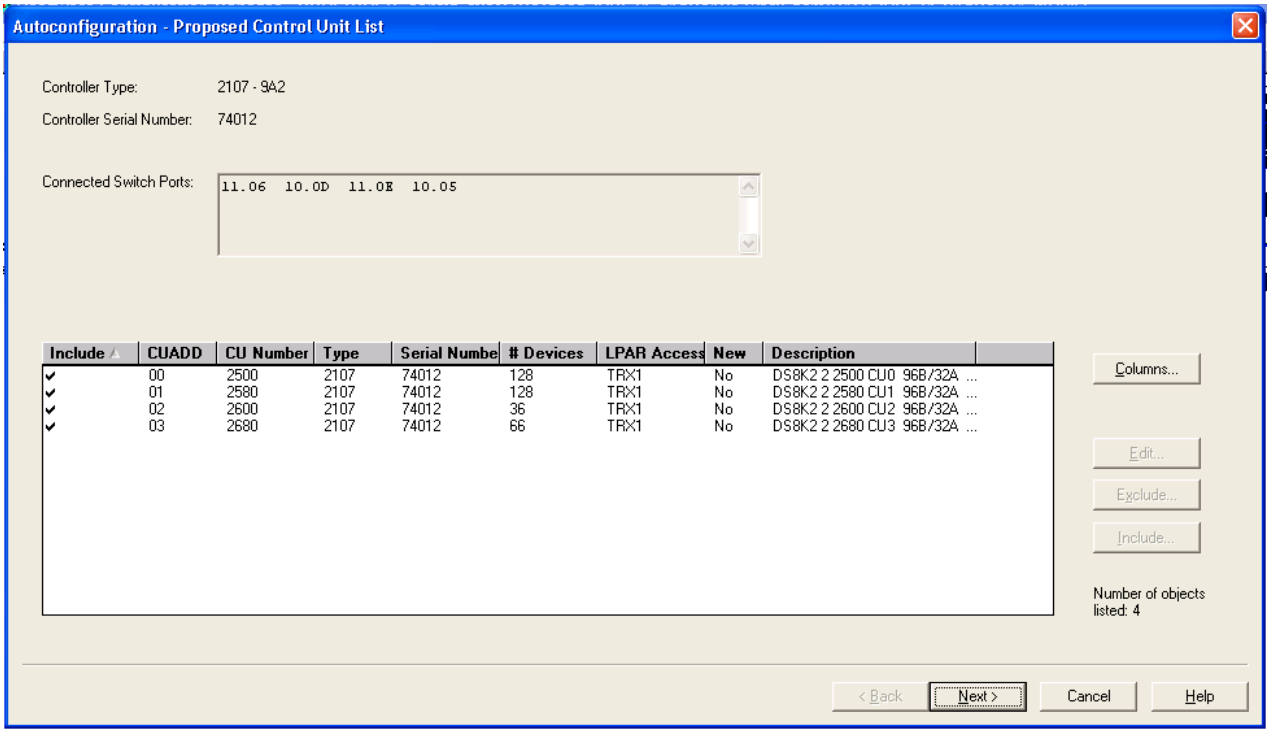


Figure 137. Proposed Control Unit List

You can include or exclude control units from definition in the IODF. You can also edit control units by pressing the **Edit...** button, with a single control unit selected.

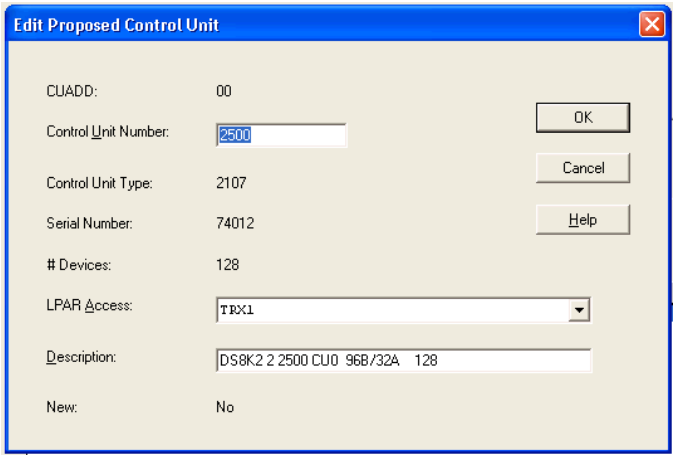


Figure 138. Edit Proposed Control Unit

Press the **Next** button on the ***Proposed Control Unit List*** to proceed to the ***Proposed Device List*** (Figure 139 on page 225).

How to work with proposed devices

The wizard page ***Proposed Device List*** is shown in attended mode only. It presents the device ranges for the one controller currently being configured.

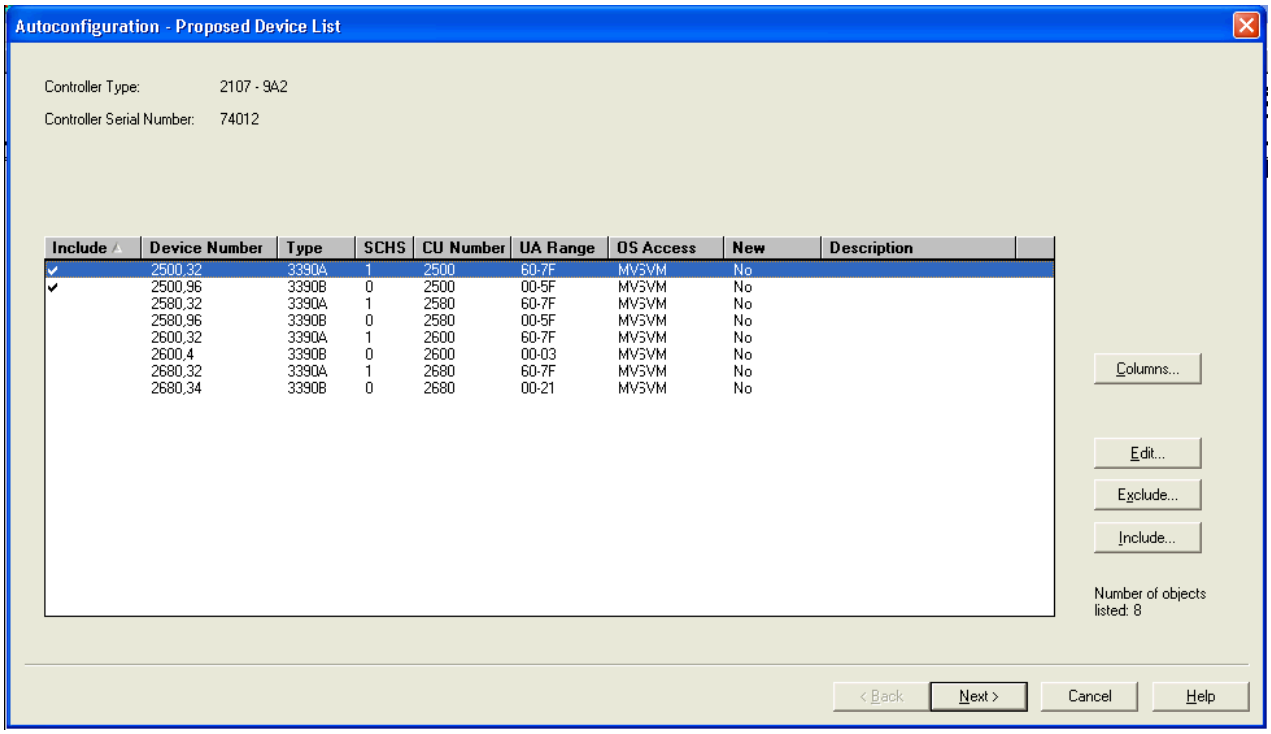


Figure 139. Proposed Device List

You can include or exclude device ranges from definition in the IODF. You can also edit device ranges by pressing the **Edit...** button, with a single range selected.

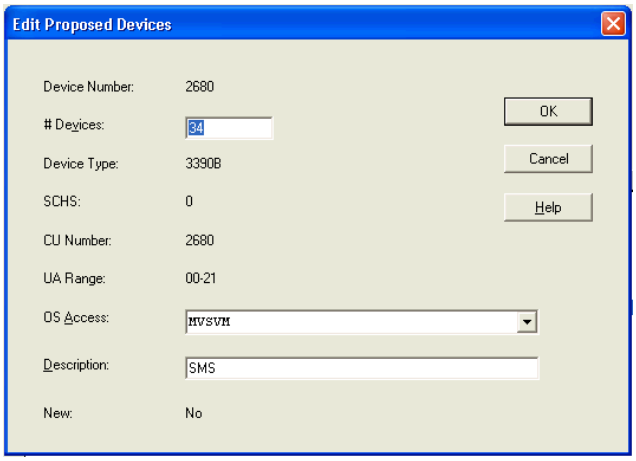


Figure 140. Edit Proposed Devices

Press the **Next** button on the the ***Proposed Device List*** to proceed to the ***Summary*** page (Figure 143 on page 228).

How to apply your own numbering scheme for PAV devices

If you work with PAV base and alias devices, you can let I/O Autoconfiguration assign device numbers according to a scheme as specified with the AUTO_SS_DEVNUM_SCHEME autoconfiguration policy. For details about the offered schemes, you can press the ***?*** buttons in the ***Autoconfiguration Policies*** dialog, or refer to the *z/OS HCD User's Guide*.

However, if you set this policy to NONE, I/O Autoconfiguration offers you dialogs to assign your own numbering to the discovered control units and devices. From the ***Discovered Controller List***, I/O Autoconfiguration first navigates to the ***Map Proposed Control Units*** (Figure 141 on page 226) where you assign your appropriate control unit numbers.

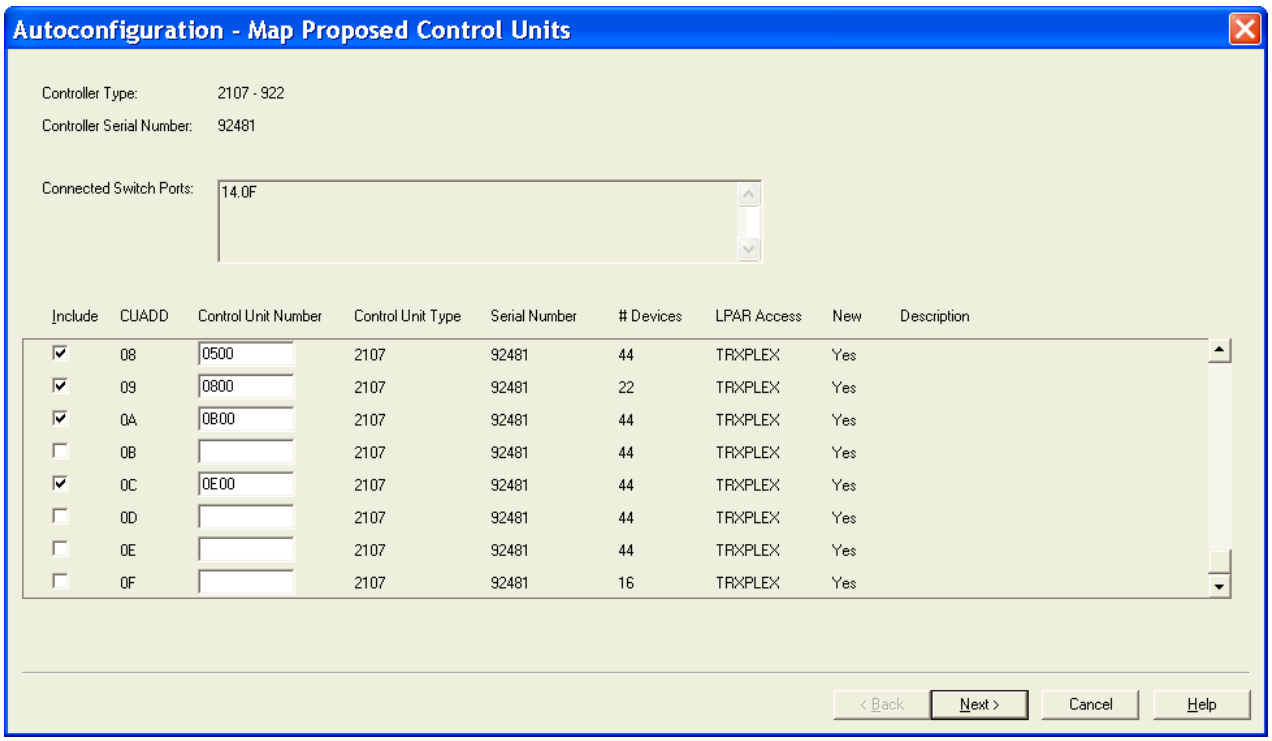


Figure 141. Map Proposed Control Units

If you work in attended mode, pressing the **Next** button navigates to the ***Proposed Control Unit List*** which reflects your assigned numbers. You have the opportunity to edit the shown values as described in "How to work with proposed control units" on page 224.

In attended mode, I/O Autoconfiguration proceeds to the ***Map Proposed Devices*** dialog (Figure 142 on page 227) where you now can assign appropriate device numbers.

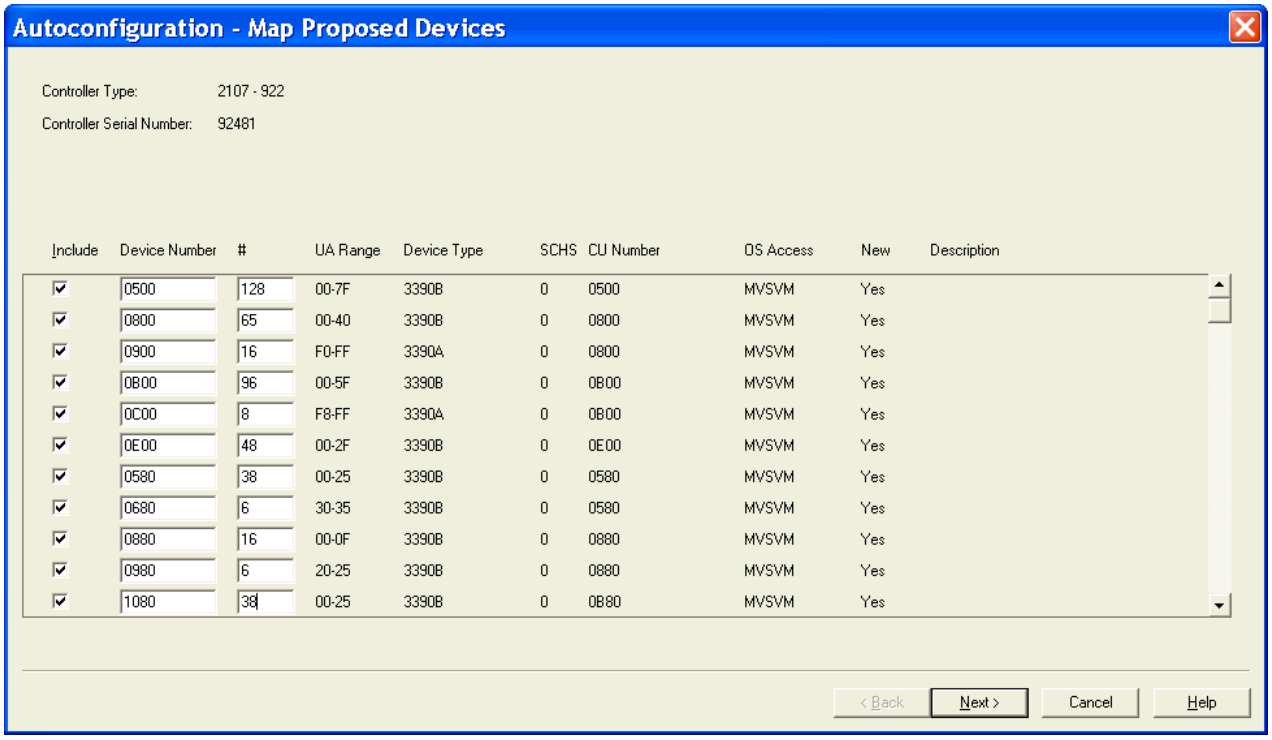


Figure 142. Map Proposed Devices

If you work in attended mode, pressing the **Next** button navigates to the ***Proposed Device List*** which reflects your assigned numbers. You have the opportunity to edit the shown values as described in "How to work with proposed devices" on page 225.

How to work on the Summary page

The ***Summary*** page is shown in both operation modes:

- In attended mode, it shows the definitions made for the current controller only, and you can reject them.
- In unattended mode, it shows the definitions made for all controllers. You cannot reject them.

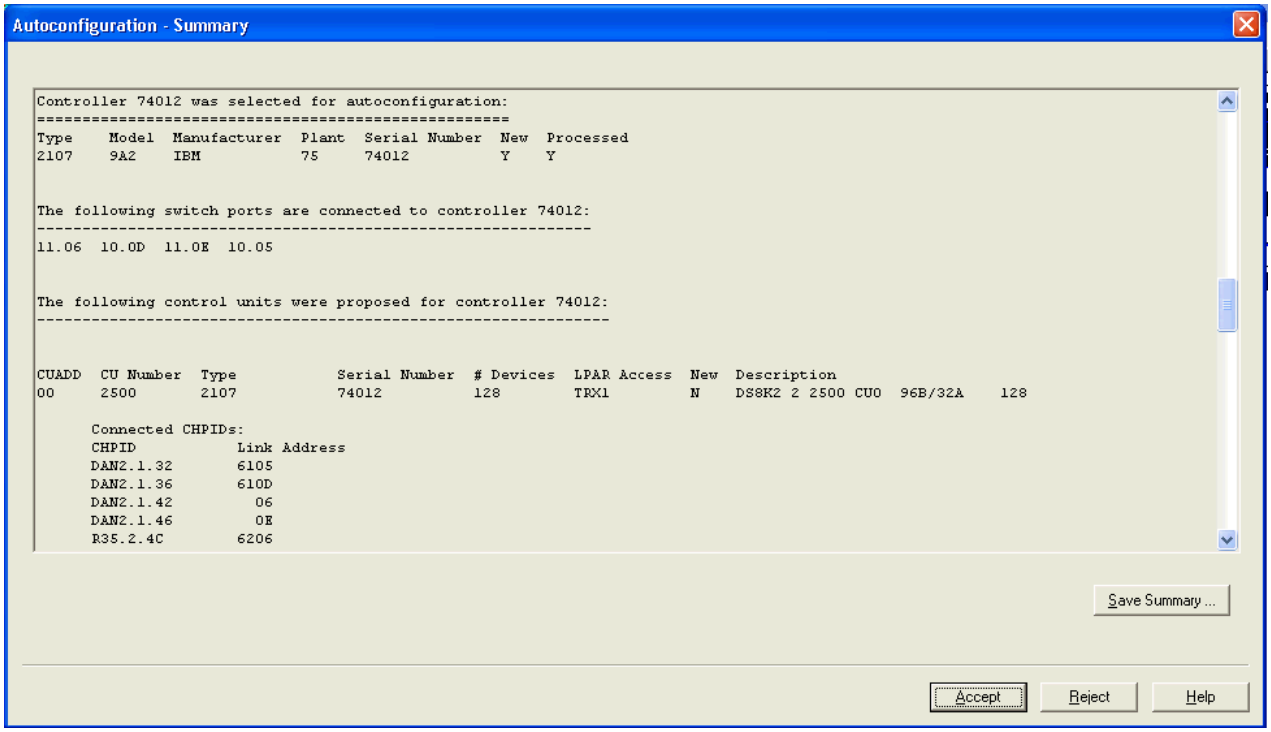


Figure 143. Autoconfiguration - Summary (excerpt)

Both buttons, ***Accept*** and ***Reject***, lead back to the ***Discovered Controller List*** (Figure 136 on page 223). The column ***Processed*** in the controller list shows which controllers were configured (and accepted). The check mark in column ***Include*** is cleared for configured controllers. In attended mode, this happens for one controller at a time.

You can now either:

- mark further controllers for autoconfiguration and press the ***Next*** button, or
- leave the wizard using the ***Close*** button.

How to define autoconfiguration policies

Many aspects of autoconfiguration are determined by user-defined policies. To edit them, press the ***Policies...*** button on the ***Autoconfiguration - Options*** page (Figure 135 on page 223). This invokes the ***Autoconfiguration Policies*** dialog shown in Figure 144 on page 229.

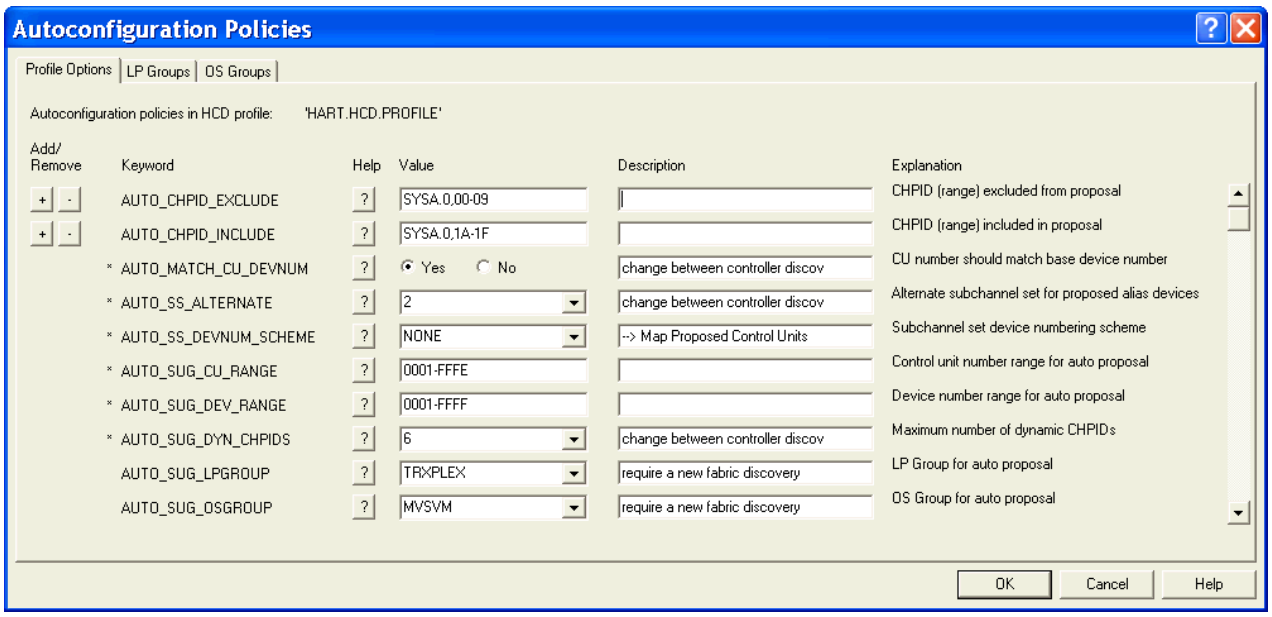


Figure 144. Autoconfiguration Policies

On the ***Profile Options*** tab of this dialog, you can edit policies represented by entries in the HCD profile. To see all available policy keywords, scroll down to the end of the list. For details on the effect of these policies, you can press the ***?*** buttons or refer to the *z/OS HCD User's Guide* for an explanation of this keyword.

For keywords that you may specify multiple times with different values, like for example, AUTO_CHPID_EXCLUDE, the dialog provides actions ***Add*** (button with the + sign) and ***Remove*** (button with the - sign) to insert or delete policy entries.

On the ***LP Groups*** tab of this dialog, you can add, edit, or delete named logical partition groups. LP groups are stored in the IODF. They determine to which partitions discovered devices are defined.

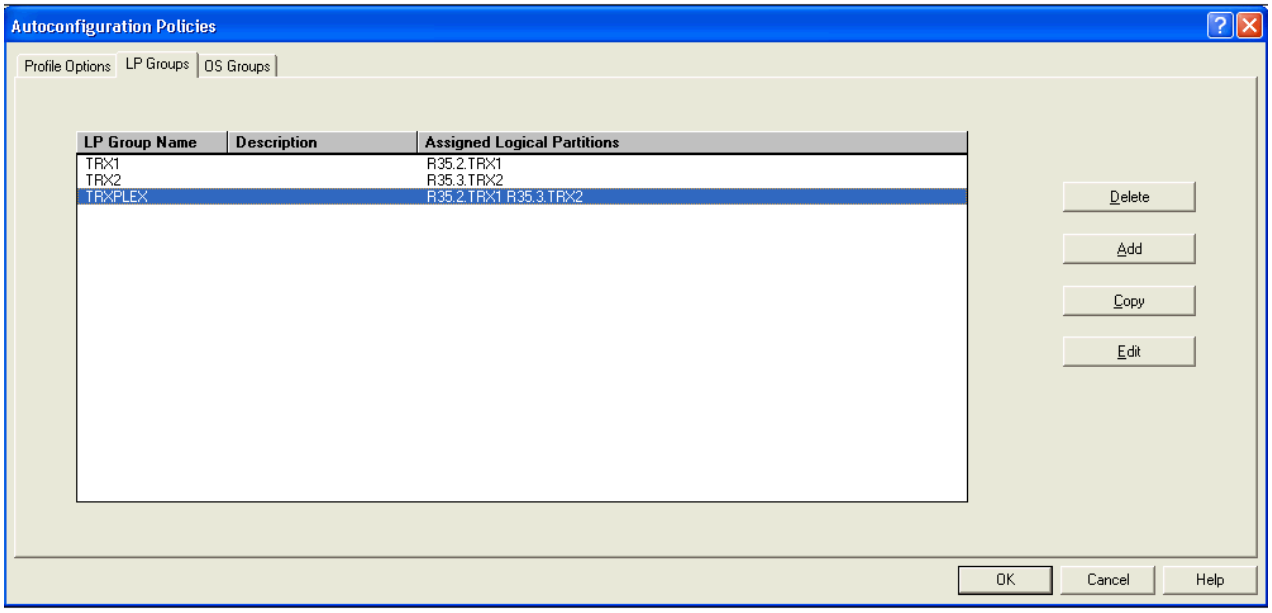


Figure 145. Autoconfiguration Policies - LP Groups

The buttons ***Add***, ***Copy*** or ***Edit*** open the respective dialogs. These dialogs all have the same layout. They allow you to assign or remove partitions from a group just by clicking the respective button. You can filter the list of unassigned partitions by processor ID.

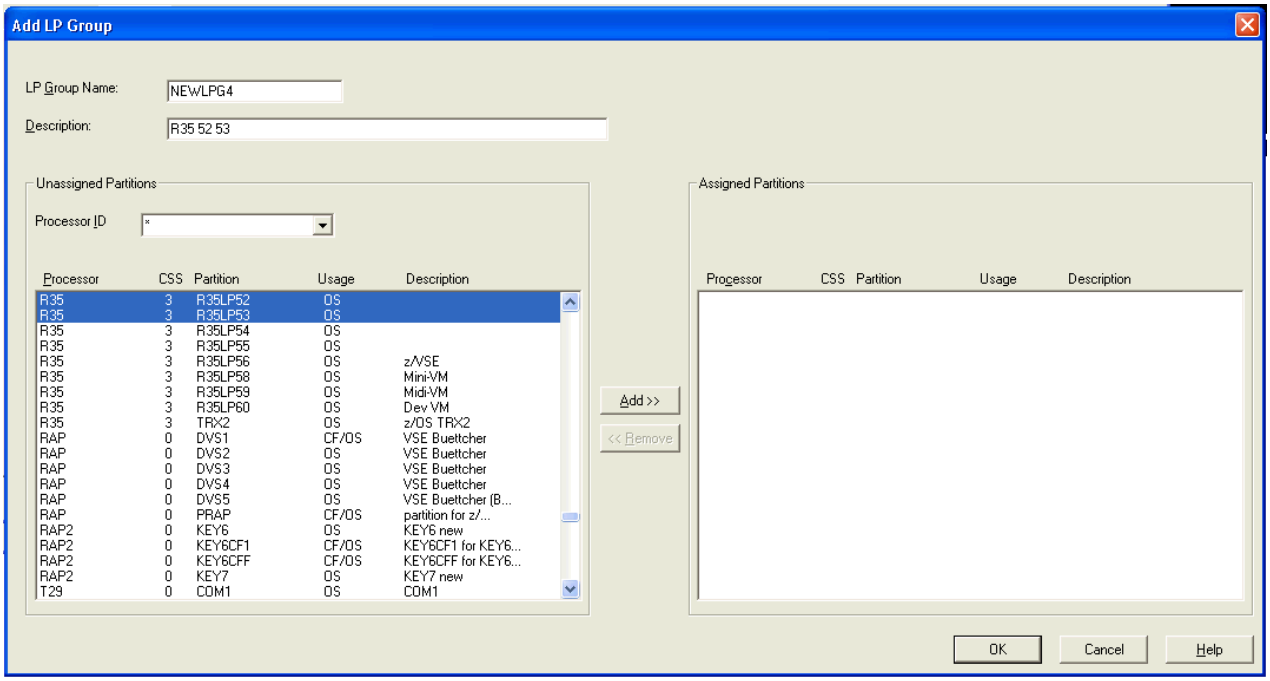


Figure 146. Add LP Group

On the ***OS Groups*** tab of this dialog, you can add, edit, or delete named operating system groups. OS groups are also stored in the IODF. They determine to which operating systems discovered devices are defined.

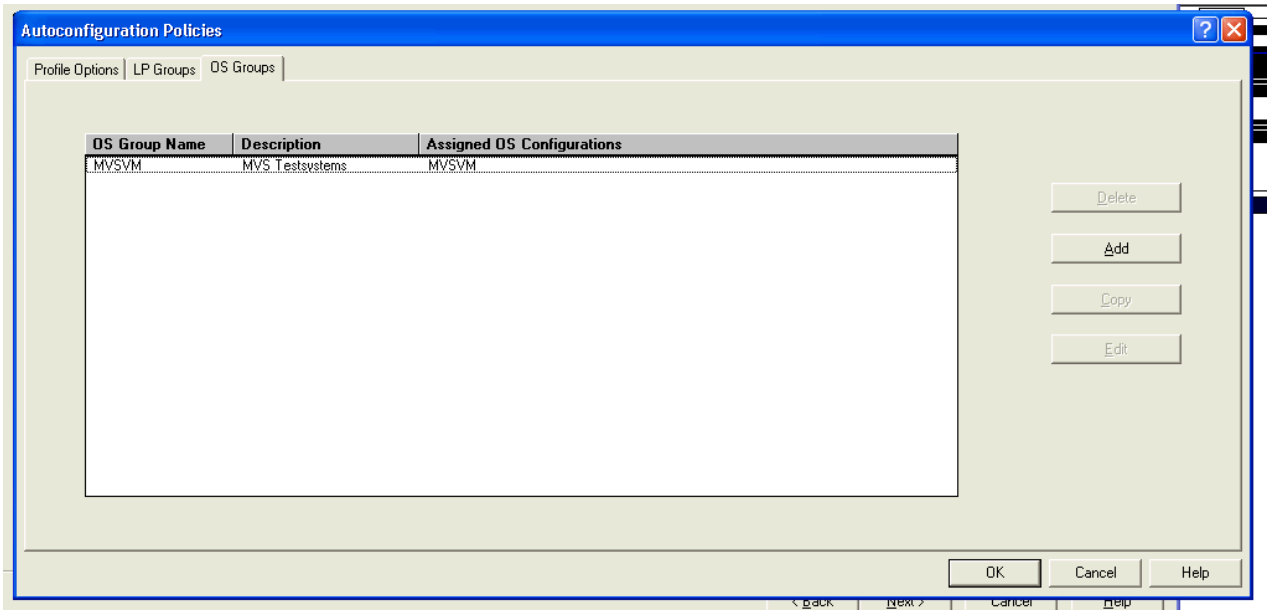


Figure 147. Autoconfiguration Policies - OS Groups

Working with OS groups is very similar to working with LP groups. Figure 148 on page 231 shows an example for editing an OS group.

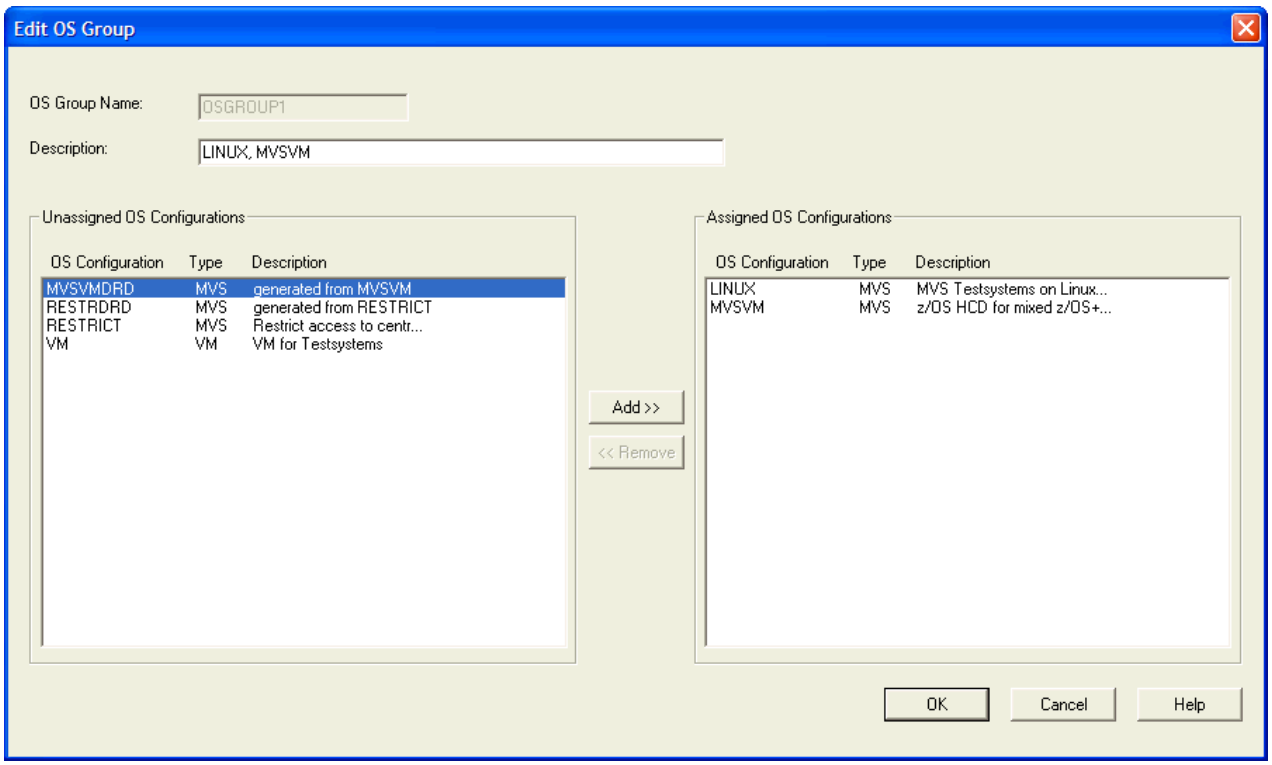


Figure 148. Edit OS Group

How changed control units may affect HCM

During I/O Autoconfiguration, HCM may change definitions of existing control units. For example, all control units of an existing controller may have their type changed from 2105 to 2107.

This does not change the ***physical label*** of the controller. For example, the controller may have had an automatically generated physical label ***2105 #1***. Even if all contained control units change their type from 2105 to 2107, the physical label of the controller remains unchanged.

This is the same behavior as performed by resynchronization. HCM can, in general, not reliably tell whether a physical label was generated or user-defined.

Also, the physical description file (PDF) associated with the controller is not changed in this case. HCM cannot reliably tell whether, for example, the *DS8000* PDF file would be more appropriate, because users may have adapted the existing physical description manually to match their needs.

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 is 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 232 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 42 for more information about physical inconsistencies.

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.

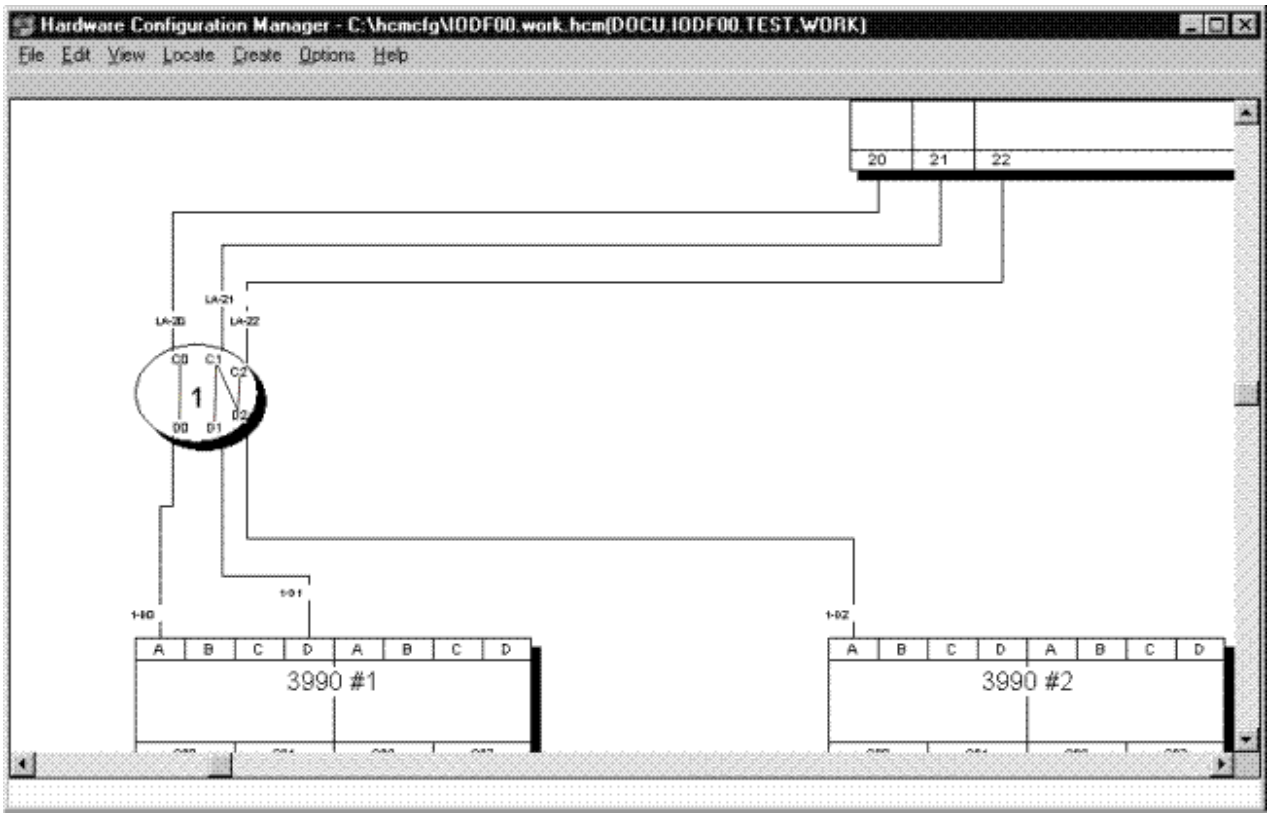


Figure 149. Example for Implied Ports

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.

Implied ports

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 231 and “Aggregating CHPIDs” on page 233 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 233 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 231 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 320 for an example of the use of this utility.

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/Copying/Editing an I/O subsystem

HCM offers utilities that allow you to create, copy or edit an I/O subsystem consisting of:

- controllers (with segments and control units) and
- strings (with devices).

If you use the **Create I/O Subsystem** or **Copy I/O Subsystem** utility, you select the controller model as the first step:

- If you select a PDF as your controller model, a new I/O subsystem is created on base of the selected controller-PDF.
- If you select a certain controller as the controller model, a new I/O subsystem is created as a copy on base of the selected controller.

Invocation

You can invoke the utilities in the following ways:

- using the **Copy I/O Subsystem...** or **Edit I/O Subsystem...** choices from a controller's context menu
- using the **I/O Subsystem...** pull-down choice from the **Create** menu bar choice

Using the utility to create an I/O subsystem

The utility guides you through the following steps:

1. **Set Controller Attributes and Subsystem Type:** Select a controller or a controller's physical description file (PDF). You can define a label prefix for the controller, and, if applicable, the number of controllers within the I/O subsystem.
2. **Add control units:** Attributes for control units consist of type, number, subsystem identifier (SSID), offset between subsequent CU numbers, serial CU number, and description.
3. **Connect control units:** Connect control units to controllers (and their segments).
4. **Define strings:** Select a string PDF (based on a selected type). Set the following attributes: number of devices inside string, starting device number, offset between subsequent device numbers, label prefix, and description.
5. **Connect string interfaces:** Connect string interfaces and controller device interfaces. Selectable interfaces are based on the settings made in steps “1” on page 235 and “4” on page 235.
6. **Connect strings:** Connect strings and control units. Selectable objects are based on the settings made in steps “3” on page 235 and “5” on page 235.

Using the utility to copy an I/O subsystem

With this utility you can also copy an existing I/O subsystem with its topology. That is, the utility duplicates the quantity and attributes of contained control units, the quantity and attributes of strings (with devices), and the connections between contained objects.

The utility guides you through the following steps:

1. **Set Controller Attributes and Subsystem Type:** Select an existing controller.

   To create a valid new I/O subsystem, you must change the CU numbers (which must be unique in a valid configuration). Also, you can change other I/O subsystem attributes (like a controller's serial number or subsystem identifier (SSID), a string's label and I/O device IDs). The utility supports you in globally changing CU numbers and I/O device IDs via specification of a new first CU number and I/O device ID. All other CU numbers and I/O device IDs are changed according to the original topology (That means, that the original gaps between the values will remain).
2. **Add control units:** Accept or change the generated new control unit numbers. The utility may issue a message on the current page if you must change possibly invalid control unit numbers before continuing.
3. The rest of the steps of the copy functionality are similar to steps “3” on page 235 through “6” on page 235 from the create functionality. However, they provide only information about the topology that the copied I/O subsystem will have, and no updates are possible.

Using the utility to edit an I/O subsystem

To ease the task of modifying an existing I/O subsystem consisting of many control units and devices, HCM offers an ***Edit I/O Subsystem*** wizard which you can invoke from a controller's context menu. This wizard lets a user perform similar actions in a single step when upgrading a controller. For example, it lets the user, in a single work unit, increase the number of control units, strings, and devices, and change the object topology inside the I/O subsystem.

This utility guides you through the same steps as the **Copy I/O Subsystem** utility, allowing you to apply the desired appropriate changes in each step.

Exporting IOCP or I/O configuration statements

In connected mode, you can use the dialog shown in Figure 150 on page 236 to create IOCP input data or I/O configuration statements for the currently accessed IODF. You can export the created output either into a data set on the host or into a file on the workstation.

Exporting IOCP input data is only available for accessed production IODFs. You need to select the processor for which you want to build the IOCP input file. Into the ***Title*** field, enter the identification information you want to be written on the first header line of the IOCP input file. Also, you need to specify whether to build the IOCP input file for stand-alone IOCP or for batch IOCP.

For exporting I/O configuration statements, you can select which type of configuration statements the data set or file should contain, namely either:

- for a processor configuration,
- for an OS configuration, or
- for a switch configuration.

Note: This HCM dialog offers the same services as the functionality implemented in HCD on the host. For detailed information required to fill the HCM dialogs, either refer to the HCM online help or to the topic ***Build an IOCP input data set*** in *z/OS HCD User's Guide*.

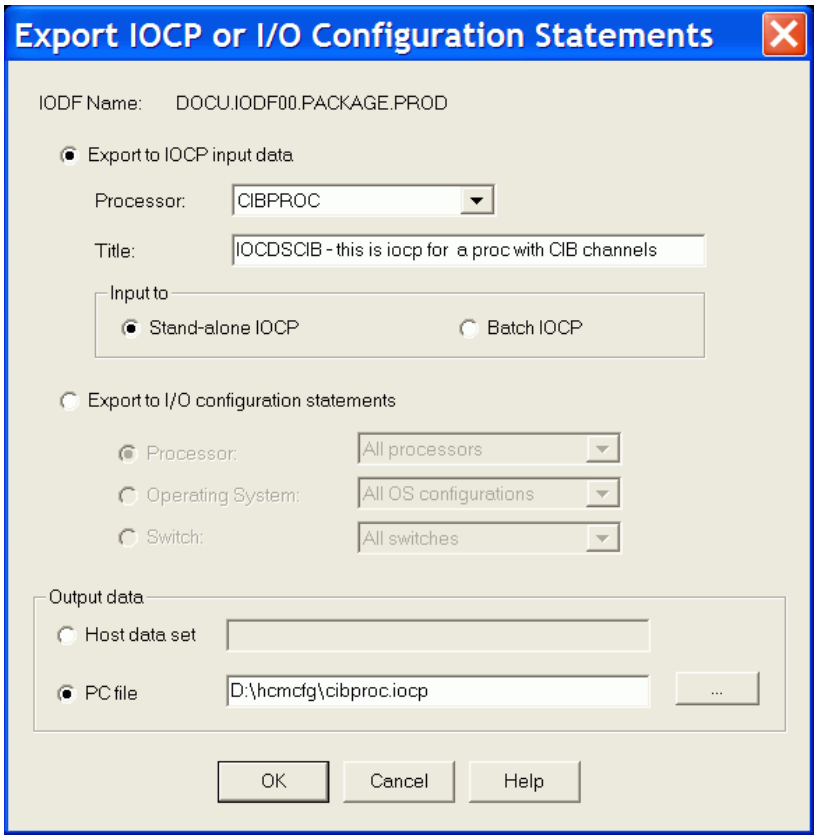


Figure 150. Export IOCP or I/O Configuration Statements

Importing I/O configuration statements

In connected mode, you can use the dialog shown in Figure 151 on page 237 to import/migrate I/O configuration statements or IOCP input data into an IODF. You can select whether the specified input data will be taken from the host or from the workstation.

To display this dialog, choose ***IODFs*** from the ***File*** menu to open the ***IODFs*** dialog. Select the work IODF to which you want to migrate your input data and click the ***Import*** button in the ***Cfg. Statements*** section.

Note:

1. This HCM dialog offers the same services as the functionality implemented in HCD on the host. For detailed information required to fill the HCM dialogs, either refer to the HCM online help or to the topic ***Migrating input data sets using the HCD dialog*** in *z/OS HCD User's Guide*.
2. This task just changes the IODF on the host without updating the associated HCM configuration file accordingly. Therefore, HCM will perform a resynchronization when opening the configuration the next time.

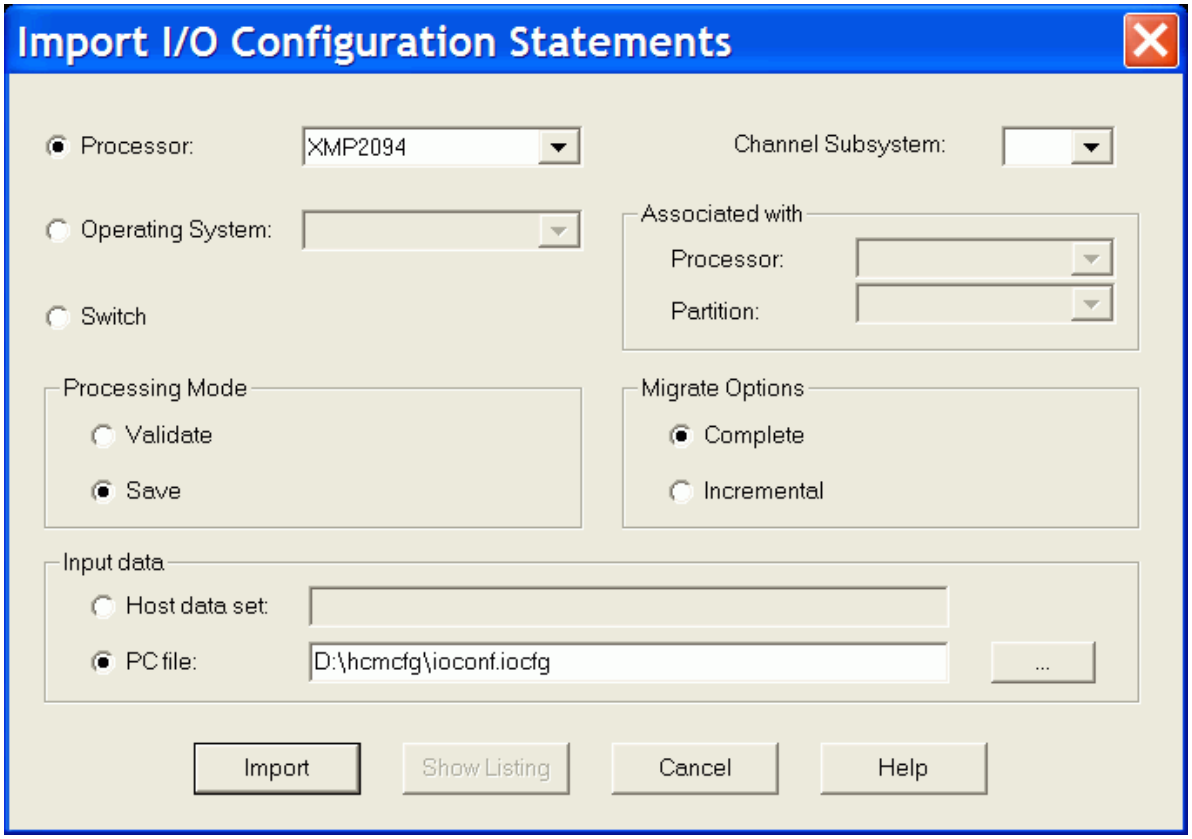


Figure 151. Import I/O Configuration Statements

Browsing performance data

This utility lets you browse performance data for objects of certain types (see "Objects that provide performance data" on page 240). You invoke this utility for a selected object with one of the methods listed below. HCM then calls your preferred internet browser to present the performance data. Note that your browser must be able to display XML documents with XSL style sheets, for example, Mozilla Firefox, Microsoft Internet Explorer 5.5 or later, or Netscape 7.0 or later.

The browser connects to the z/OS RMF Distributed Data Server (DDS) which in turn provides the collected performance data of the object under focus.

Using this utility may help you to ensure a better performance-related operation of your system. For example, you can easily track down which other objects are affected by a certain object that you are currently examining because of its bad performance. Thus, HCM can help you to easier inspect performance bottlenecks.

You need to observe the following **prerequisites and restrictions** when using this utility:

- The z/OS RMF Distributed Data Server (DDS) must be installed, set up and running in the sysplex. For information about how to exploit the DDS, refer to in the .
- Since the DDS is only available on z/OS, this utility is not available for z/VM.
- Performance data is only available for objects that belong to the sysplex of the system to which the user has logged on.

  Note: It is recommended to focus your current view to the active sysplex: Open the ***Change View Attributes*** dialog (see "Filtering your view" on page 33) and click the **Active Sysplex** check box.
- The loaded HCM configuration file/IODF should describe the current sysplex comprising the system to which you logged on.
- You cannot browse performance data in stand-alone mode.

You have the following alternatives for accessing an object's performance data:

Alternative 1:

You can directly access performance data of a selected object as described in the following:

- Use the **Performance...** choice from an object's context menu where applicable, for example, from a processor's context menu.
- Use the **Performance...** button from an object's dialog where applicable, for example, from the **CHPID** dialog.

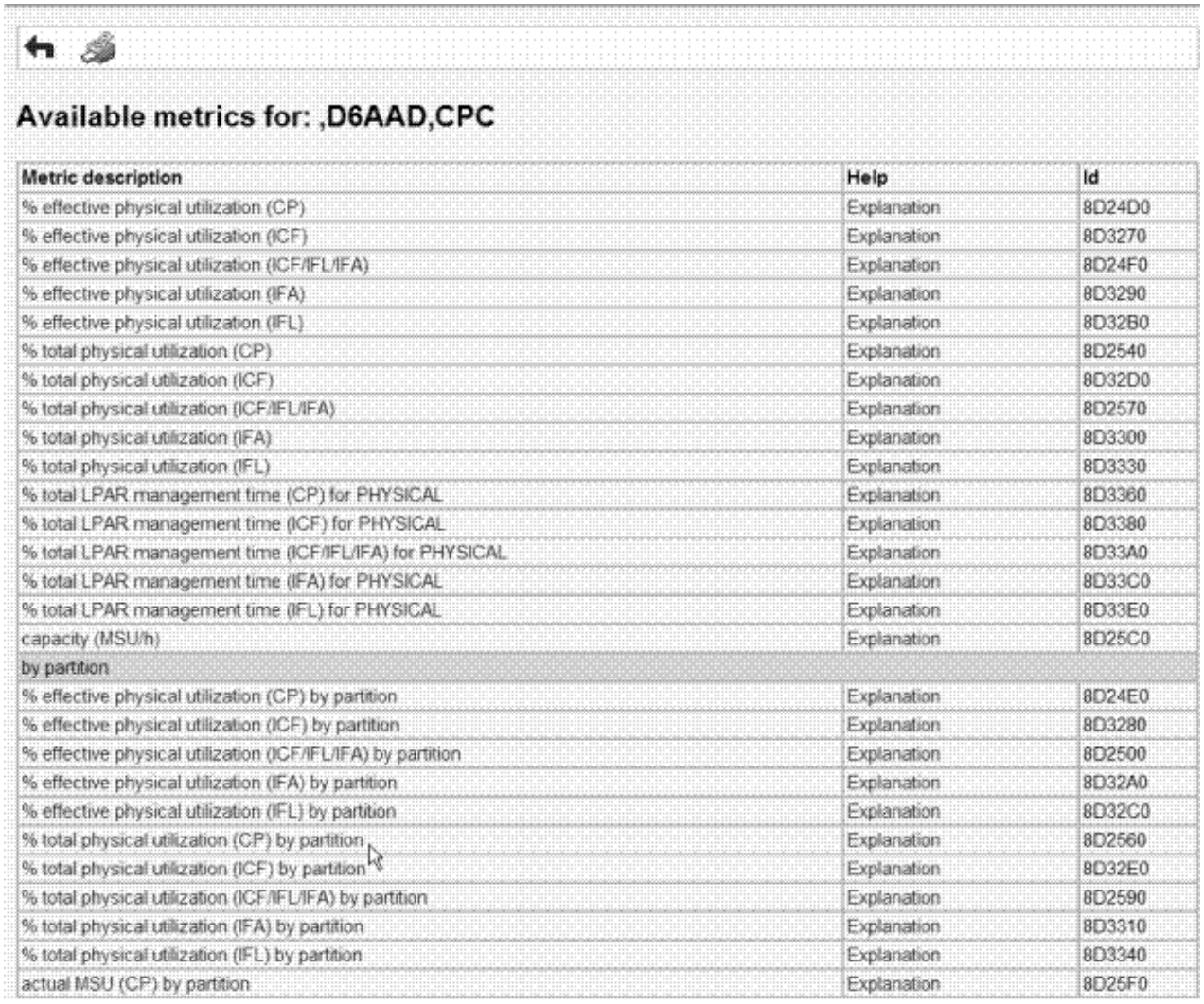
However, from the opened browser windows, you cannot navigate to child resources.

Alternative 2:

You can also use the **Browse Performance Data...** pull-down choice of the **Utilities** menu bar choice. This choice opens the web browser which offers the following navigation features:

- You can retrieve performance metrics for the sysplex as a whole.
- You can step down the hierarchy of objects in the sysplex and retrieve metrics for available child objects of the sysplex, for example, the capacity or free space for a certain I/O device (volume) defined for a certain operating system (MVS image), or the percentage of used capacity for a logical partition. In the row that shows the resource object that you want to examine, click on **Metrics**.
- On all browser windows in this hierarchy, you can also view attributes of the listed objects by clicking on the entry in column **Attributes** for an object.

As an example, let us assume you want to view performance data of a 2094 processor in your sysplex: Open the **Processor** dialog for this processor and press the **Performance...** button. A browser window showing the available metrics for this processor opens:



Available metrics for: ,D6AAD,CPC

Metric description	Help	Id
% effective physical utilization (CP)	Explanation	8D24D0
% effective physical utilization (ICF)	Explanation	8D3270
% effective physical utilization (ICF/IFL/IFA)	Explanation	8D24F0
% effective physical utilization (IFA)	Explanation	8D3290
% effective physical utilization (IFL)	Explanation	8D32B0
% total physical utilization (CP)	Explanation	8D2540
% total physical utilization (ICF)	Explanation	8D32D0
% total physical utilization (ICF/IFL/IFA)	Explanation	8D2570
% total physical utilization (IFA)	Explanation	8D3300
% total physical utilization (IFL)	Explanation	8D3330
% total LPAR management time (CP) for PHYSICAL	Explanation	8D3360
% total LPAR management time (ICF) for PHYSICAL	Explanation	8D3380
% total LPAR management time (ICF/IFL/IFA) for PHYSICAL	Explanation	8D33A0
% total LPAR management time (IFA) for PHYSICAL	Explanation	8D33C0
% total LPAR management time (IFL) for PHYSICAL	Explanation	8D33E0
capacity (MSU/h)	Explanation	8D25C0
by partition		
% effective physical utilization (CP) by partition	Explanation	8D24E0
% effective physical utilization (ICF) by partition	Explanation	8D3280
% effective physical utilization (ICF/IFL/IFA) by partition	Explanation	8D2500
% effective physical utilization (IFA) by partition	Explanation	8D32A0
% effective physical utilization (IFL) by partition	Explanation	8D32C0
% total physical utilization (CP) by partition	Explanation	8D2560
% total physical utilization (ICF) by partition	Explanation	8D32E0
% total physical utilization (ICF/IFL/IFA) by partition	Explanation	8D2590
% total physical utilization (IFA) by partition	Explanation	8D3310
% total physical utilization (IFL) by partition	Explanation	8D3340
actual MSU (CP) by partition	Explanation	8D25F0

Figure 152. Available performance data for a processor

If you want to understand the meaning of a metric, click on ***Explanation*** in the **Help** column.

If you want to know the total utilization of the physical standard processors by the partitions in the CPC, click on *% total physical utilization (CP) by partition*. Clicking on this metric opens another view presenting the desired results, in our example, a list of partitions, showing the percentage of utilization for each (Figure 153 on page 240).

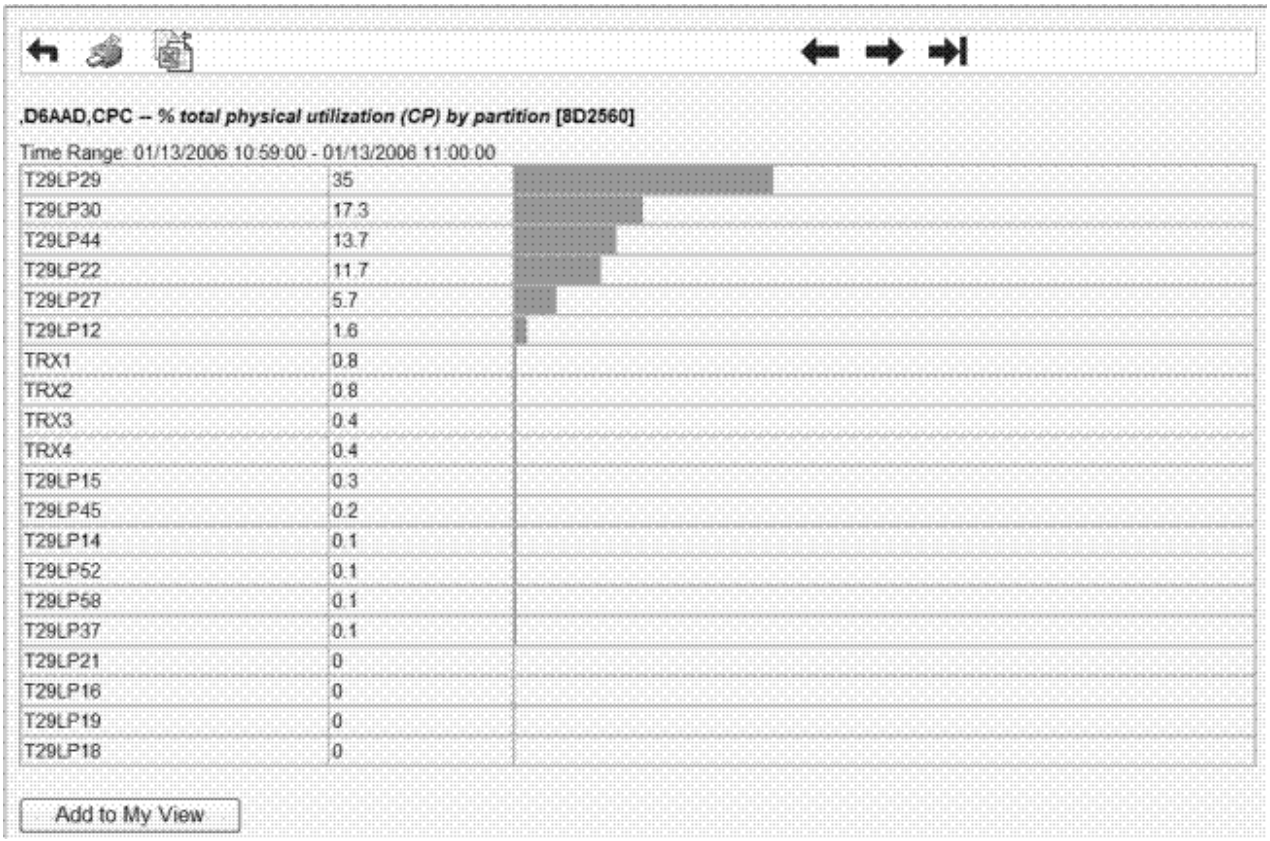


Figure 153. Total utilization of the processors by partitions

Note: If you run into problems when trying to access this utility, look for help in "Problems when using the RMF DDS" on page 389.

Objects that provide performance data

HCM can show performance data for the following objects. Depending on the DDS setup, a TSO user ID and password is required to access the RMF data. Otherwise, no further user interaction is necessary.

- **Processor:**

  HCM displays information about CPCs and their contained LPARs.

- **Partition:**

  Partition metrics may be related to the partition as a hardware object, or to the OS image running in the current partition. Therefore, you must choose the category for which you want the performance data. For this purpose, HCM displays the **Partition Performance Data** dialog to let you specify the RMF resource type and category for which performance data shall be displayed.

  In the example of Figure 154 on page 241, the user wants to get metrics from all volumes defined to the OS running in the current partition.

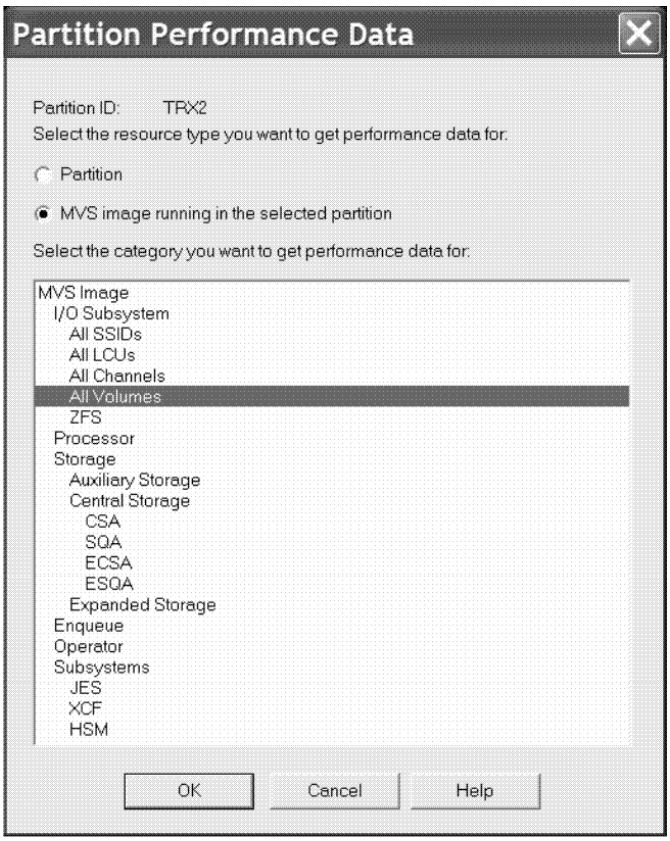


Figure 154. Partition performance data

- **CHPID:**

  A CHPID may be accessible by one or more partitions as defined in their access and candidate lists. If you want to see metrics for a selected CHPID, you must specify one of these partitions. For this purpose, HCM displays the **Partition Related Performance Data** dialog. If only one partition can access the current CHPID, this dialog is omitted.

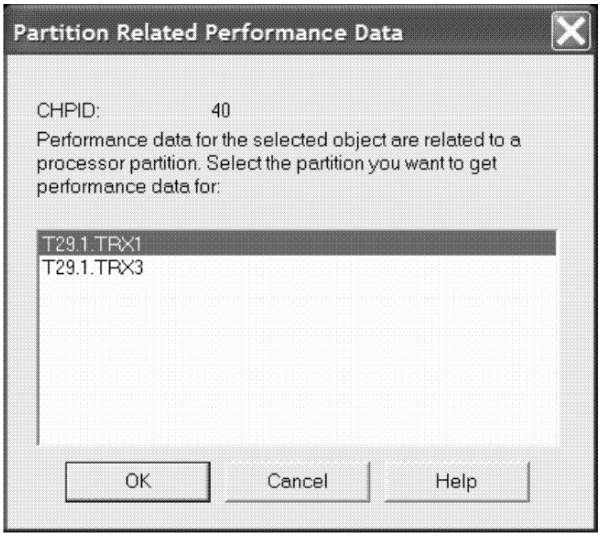


Figure 155. Partition related performance data

- **Control Unit:**

  RMF has two different resource types that correspond to a control unit: a subsystem identifier (SSID) identifies the control unit as a hardware object and a logical control unit (LCU) identifies it to the operating system. HCM lets you specify which aspect of a control unit you want to monitor:

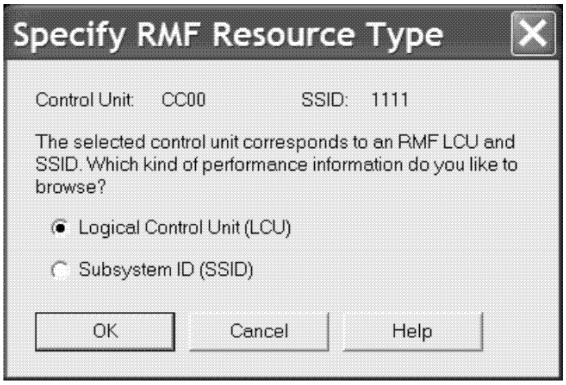


Figure 156. Specify RMF resource type

If no SSID is specified for the control unit, this dialog is omitted.

As a control unit may be related to more than one partition, you need to select one of these. For this purpose, as for CHPIDs, HCM displays the **Partition Related Performance Data** (see Figure 155 on page 241). If only one partition can access the current control unit, this dialog is omitted.

- **Devices:**

  HCM provides performance data for DASD volumes only.

  If more than one partition can access the device, HCM displays the **Partition Related Performance Data** dialog to let you select one of these (see Figure 155 on page 241).

Using the WWPN Prediction Tool Support

This utility allows you to export the FCP channel/device specific data of the I/O configuration from the currently accessed IODF into a file with comma-separated value (CSV) output format. The generated file can be used as input to the WWPN Prediction Tool to assign world-wide port names to virtualized FCP ports for configuring SAN devices.

You can invoke the ***WWPN Prediction Tool Support*** utility from the **Utilities** menu or from the ***Edit Processor*** dialog pressing the ***WPT...*** button.

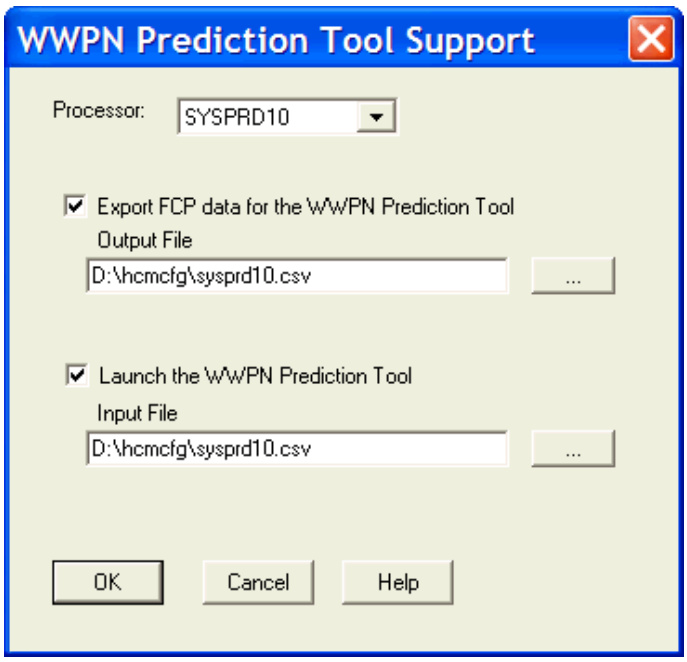


Figure 157. WWPN Prediction Tool Support

The utility helps you to perform the following tasks:

1. Export FCP data from a selected processor into a specified output file
2. Launch the WWPN Prediction Tool using a specified input file which is usually the one that you created in step “1” on page 243. This task is only enabled if the WWPN Prediction Tool is installed on your workstation.

After launching, the WWPN Prediction Tool can be used to assign WWPN values to paths between CHPIDs and devices. It returns:

- The .CSV file, updated with WWPNs, which can be imported by an SAN Manager to configure SAN devices in a fabric.
- A binary definition file, which can be transferred to the support element to be stored into the HSA for the next IML.

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 alternative 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*

© Copyright IBM Corp. 1995, 2021

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.

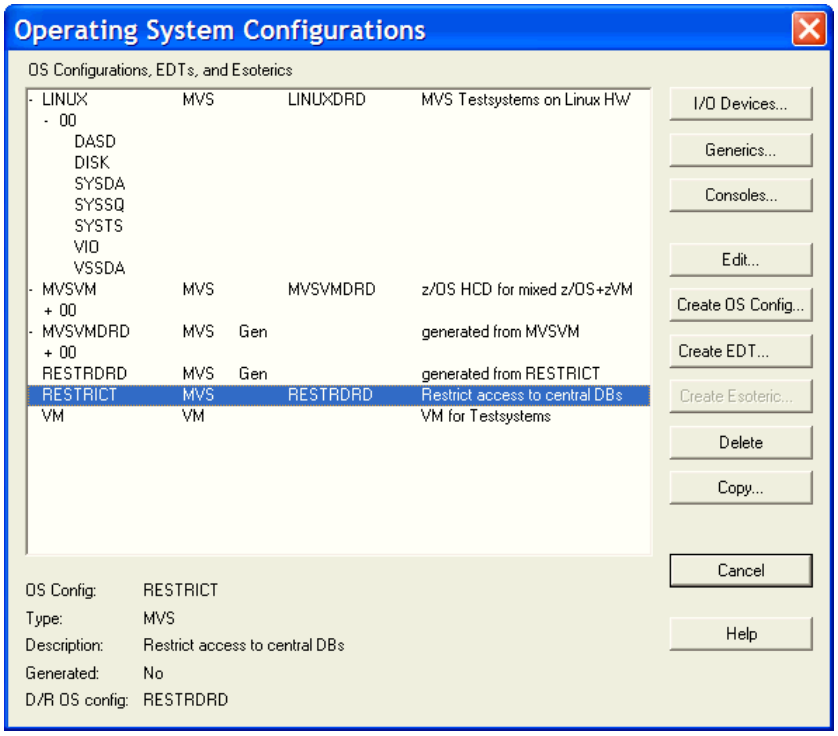


Figure 158. Operating System Configurations dialog

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 appears 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.

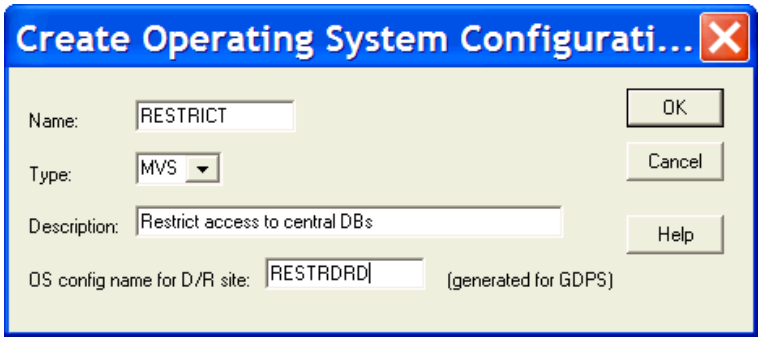


Figure 159. Create Operating System Configuration

D/R site OS configurations

You can optionally type the name of a disaster recovery (D/R) site operating system configuration into the ***OS config name for D/R site*** field. You need a D/R site OS configuration in a GDPS managed environment, where storage devices are mirrored over peer-to-peer remote copy (PPRC) connections in order to have a backup system defined for an emergency situation.

During the *Build production IODF* task, the named D/R site OS configuration is automatically created as copy of the primary OS configuration. The resulting OS configuration is called ***generated*** OS configuration.

It has all DASD devices, that are classified by PPRC usage type DUPLEX, attached with reversed OFFLINE parameter value. That is, if the primary OS configuration specifies OFFLINE=NO, the D/R site OS configuration attaches the devices with OFFLINE=YES and vice versa.

Following rules apply to ***generated*** D/R site OS configurations:

- A ***generated*** OS configuration is re-created with every *Build production IODF* from its primary OS configuration.
- With any kind of changes done to a D/R site OS configuration, (for example, modifying device-to-operating-system-relations, or changing EDT definitions), it loses its ***generated*** status. It is not automatically re-created with the next *Build production IODF*, so that the user modifications are preserved. To let it generate again from the primary site OS configuration, you must delete the modified D/R site OS configuration first.

  However, changing the description of a D/R site OS configuration does not change its ***generated*** status.
- You cannot specify the name of a D/R site OS configuration for a ***generated*** OS configuration.

The ***generated*** status of an OS configuration is indicated in the ***Operating System Configurations*** dialog (Figure 158 on page 246).

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**.

Copying an OS configuration

HCM offers a **Copy Operating System Configuration** wizard to let you copy a source operating system configuration to a new target operating system configuration. All EDT definitions (and their esoterics) of the source and all device connections are copied to the target OS configuration.

You can invoke this wizard from the **Copy...** button in the **Operating System Configurations** dialog (which you reach by selecting **OS Configurations...** from the **Edit** menu bar choice).

You only need to specify the name and optionally a description for the target OS configuration. You cannot change the type of the source OS configuration.

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. 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 group with a plus (+) sign 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 about a device group line applies only to the first device in the group. Other devices in the group may have column values that differ from the first device. To see the column values for individual devices, double-click on the plus sign.

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 z/OS operating systems are DYNAMIC and OFFLINE, those displayed for z/VM operating systems are 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).

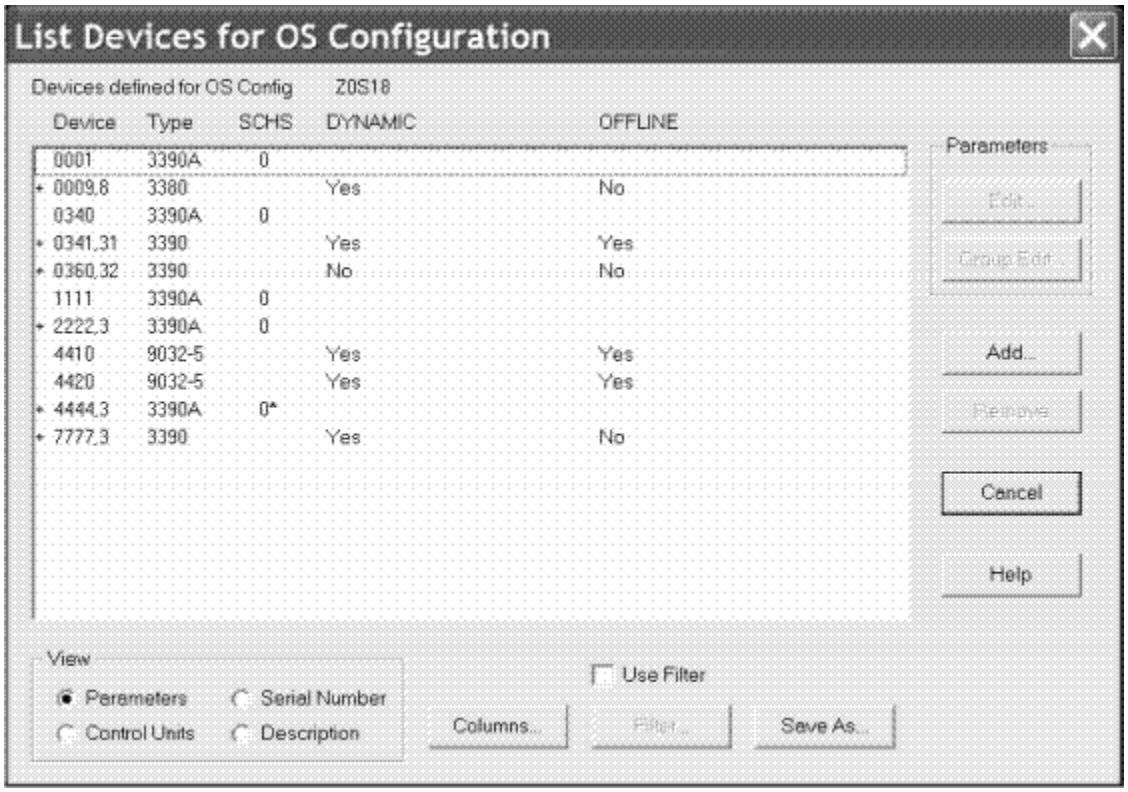


Figure 160. List Devices for OS Configuration dialog

Note:

1. 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.
2. An '*' in column **SCHS** behind the subchannel set ID for a group of devices designates that not all devices of this group are defined in the same subchannel set for the current operating system. Double - click on the plus (+) sign in front of the device group to see the settings for the individual devices.

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 about a device group line applies only to the first device in the group. Therefore, 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 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 250.

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 247 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.

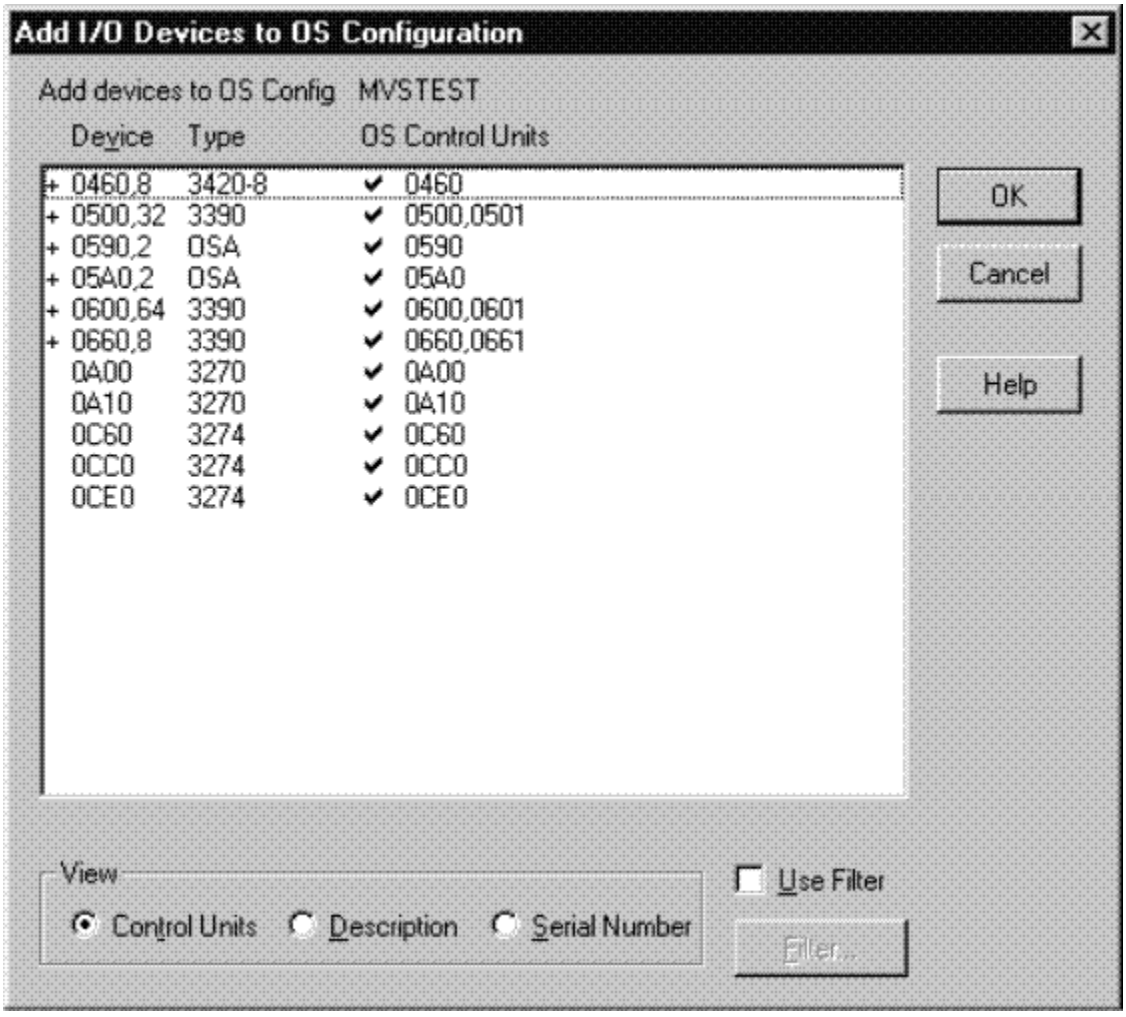


Figure 161. Add I/O Devices to OS Configuration dialog

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 displays 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 OS Configuration** dialog (Figure 160 on page 248), and press either the **Edit...** or **Group Edit...** button to open the **Define Device <=> OS Configuration Parameters** dialog (Figure 162 on page 250). You can then modify the parameters which define the selected device(s) to the OS configuration. Note that you can use the **Group Edit...** button only for selected devices of the same device type. HCM displays an error message if the selected devices are of different types, because they would be incompatible for a group edit. Also note that an '*' left to a parameter name indicates that this parameter value is not the same for all devices in a group.

If you select more than one device and press **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.

The checkboxes in column **Update** indicate whether the related parameter is updated or not when you press the **OK** button. If the checkbox is selected, then this parameter gets updated for all selected devices. This allows you to select only those parameters which you want to update for a single device or a device group, while leaving all other parameters unchanged. For a device group, all devices get the same updated value, even if the parameter value had not been the same for all devices before (indicated by an *).

Column **Update** is not visible when the devices are initially added to the OS configuration.

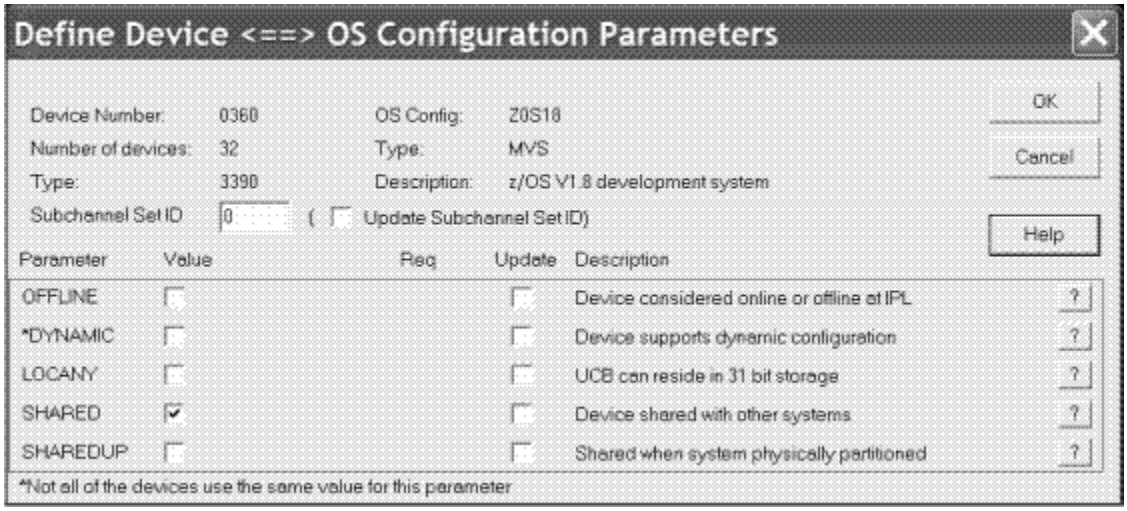


Figure 162. Define Device <=> OS Configuration Parameters dialog

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 248 or "Selecting OS configurations to which to define devices" on page 251 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.

In the field ***Subchannel Set ID*** you must specify the ID of the subchannel set where you want to put the devices. You must set this field to '0' for all devices that are not eligible for alternative subchannel sets.

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.

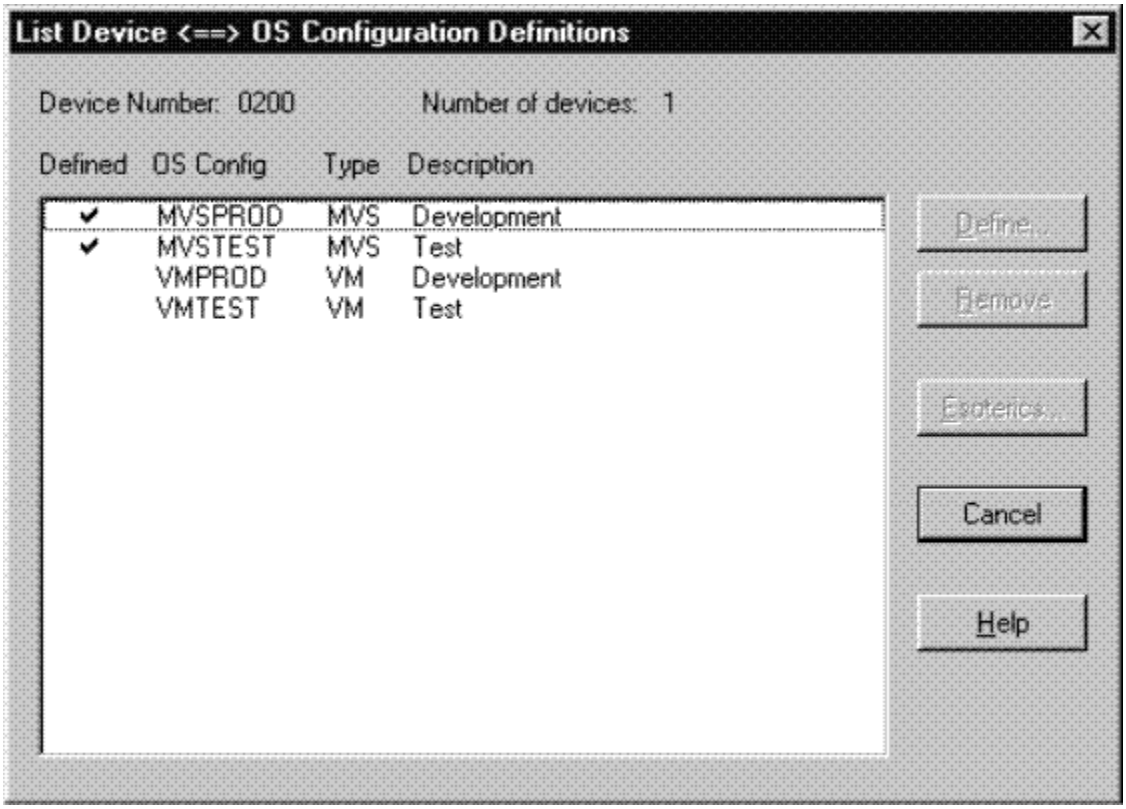


Figure 163. List Device <=> OS Configuration Definitions dialog

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 250 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 displays 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.

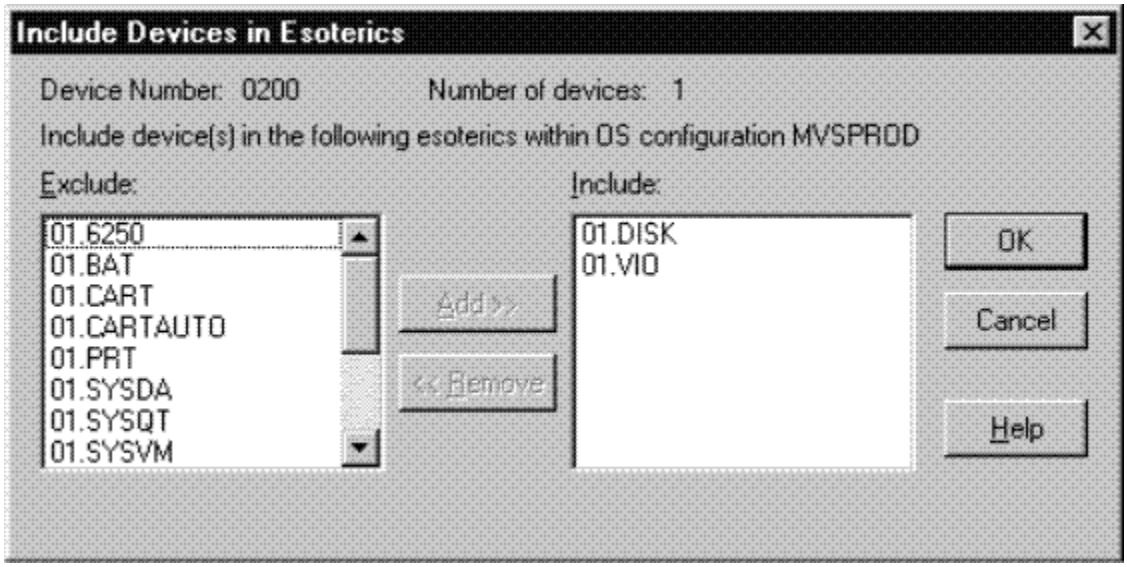


Figure 164. Include Devices in Esoterics dialog

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.

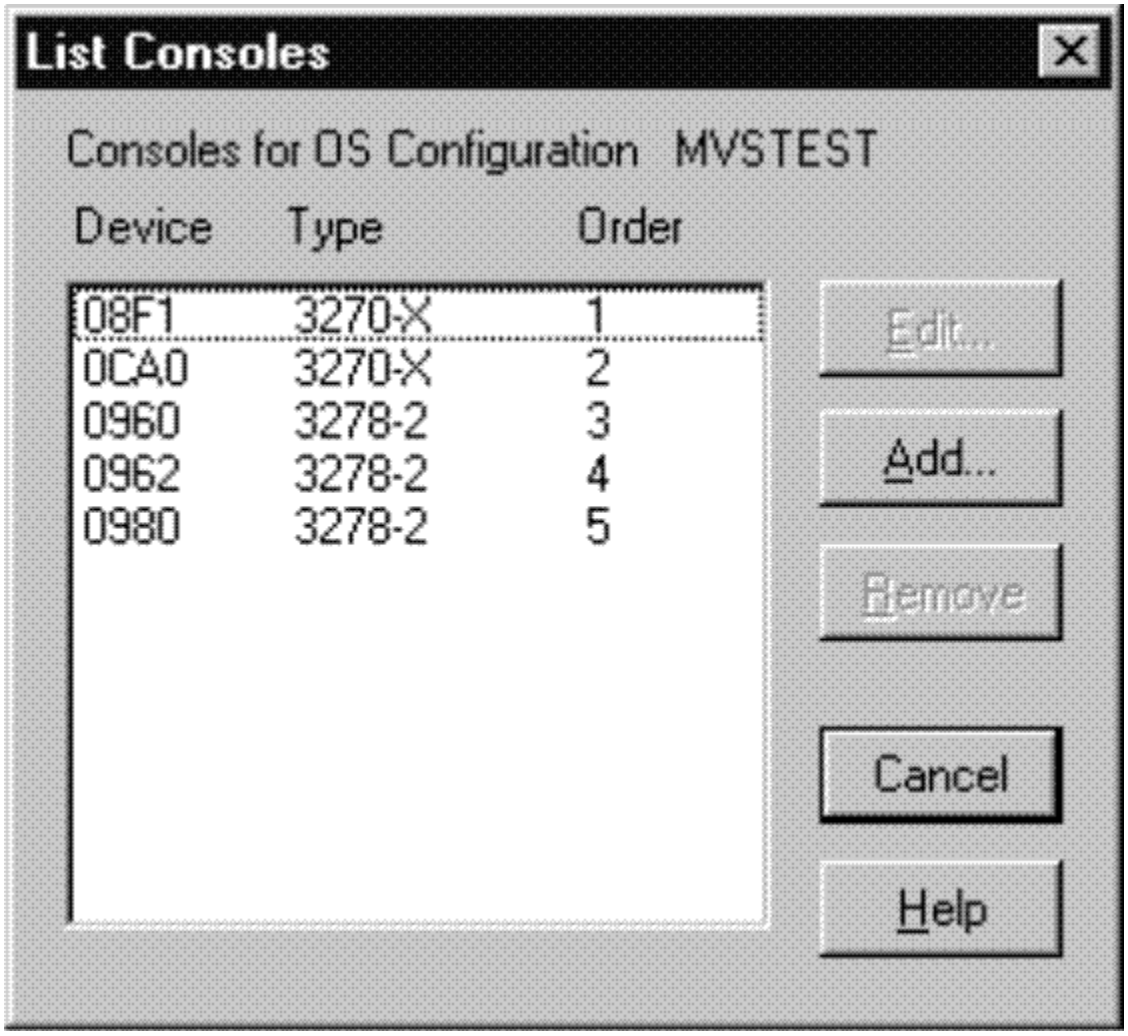


Figure 165. List Consoles dialog

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.

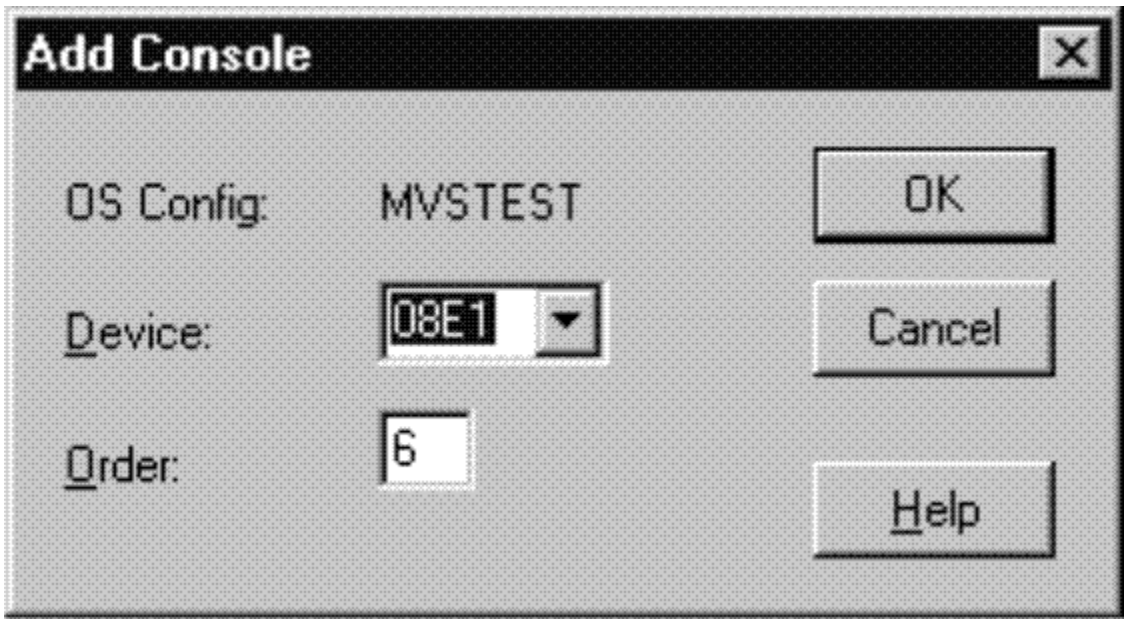


Figure 166. Add Console dialog

Removing a console from an OS configuration

Select the console and press the **Remove** button in the **List Consoles** dialog. HCM displays a confirmation dialog, then removes 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.

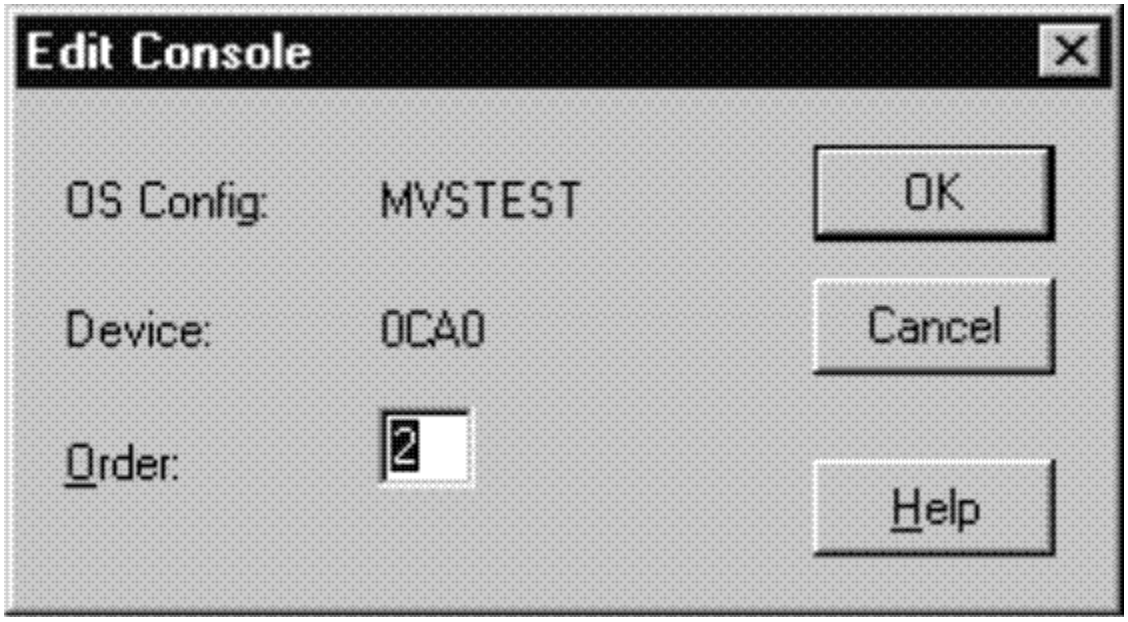


Figure 167. Edit Console dialog

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.

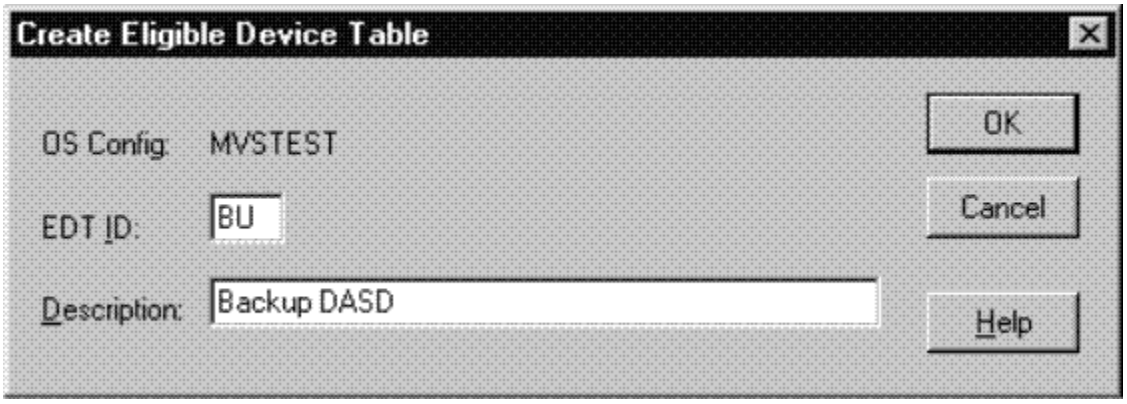


Figure 168. Create Eligible Device Table dialog

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.

Copying an EDT

HCM offers a **Copy Eligible Device Table** wizard to let you copy an EDT. All esoteric device group definitions and all device connections of the source EDT are also copied to the target EDT. The target EDT must not yet exist. It may be a member of the same or of a different OS configuration as the source EDT.

To invoke this wizard, open the **Operating System Configurations** dialog (which you reach by selecting **OS Configurations...** from the **Edit** menu bar choice). Then select an EDT and press the **Copy...** button.

To create the target EDT, the wizard queries the required information, which is basically the same as when creating a new EDT:

- the OS configuration to which it shall belong
- the target EDT ID
- and optionally a description.

Deleting an EDT

Select the EDT in the list box and press the **Delete** button in the **Operating System Configurations** dialog. HCM will displays 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.

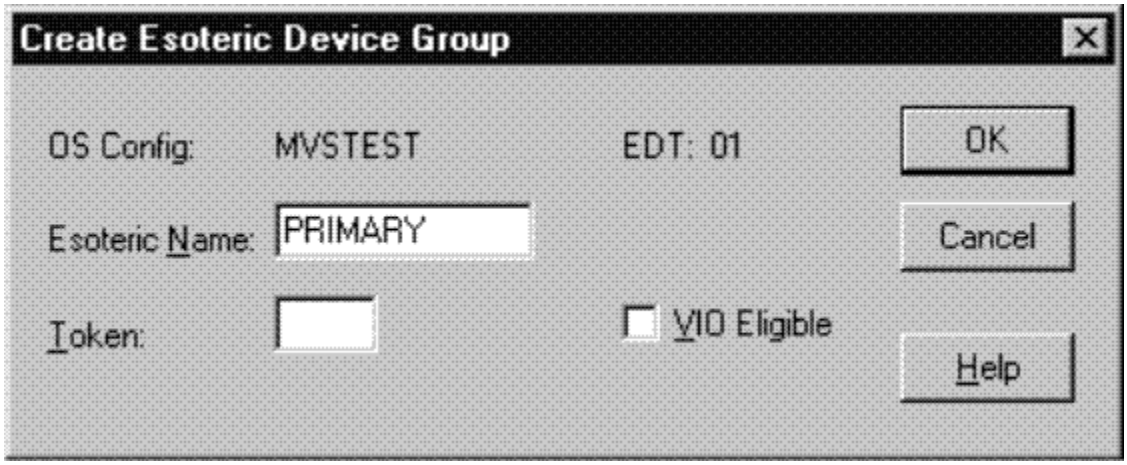


Figure 169. Create Esoteric Device Group dialog

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.

Copying an esoteric

HCM offers a **Copy Esoteric Device Group** wizard. The target esoteric must not yet exist. It may be a member of the same or of a different EDT as the source.

The devices and device connections of the source esoteric are also copied.

To invoke this wizard, open the **Operating System Configurations** dialog (which you reach by selecting **OS Configurations...** from the **Edit** menu bar choice). Then select an esoteric device group and press the **Copy...** button.

To create the target esoteric, the wizard queries the following attributes, which are basically the same as when creating a new esoteric device group (see also “Creating an esoteric” on page 256):

- the name of the esoteric device group
- the esoteric token
- and whether the VIO eligibility should be set to YES.

Deleting an esoteric

Select the esoteric in the list box and press the **Delete** button in the **Operating System Configurations** dialog. HCM displays 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.

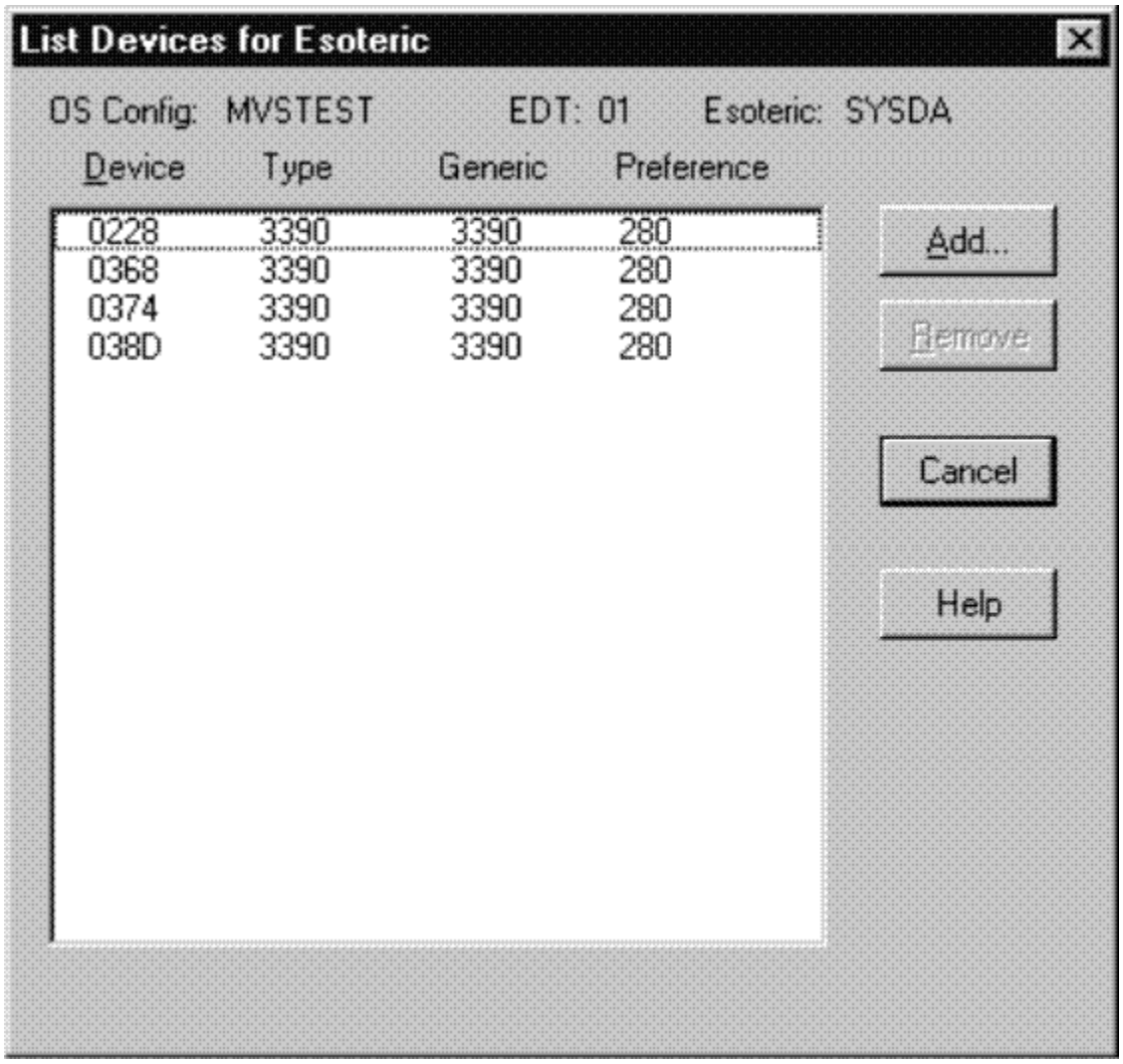


Figure 170. List Devices for Esoteric dialog

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 about a device group line applies only to the first device in the group. Therefore, 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 247 for more details on filtering).

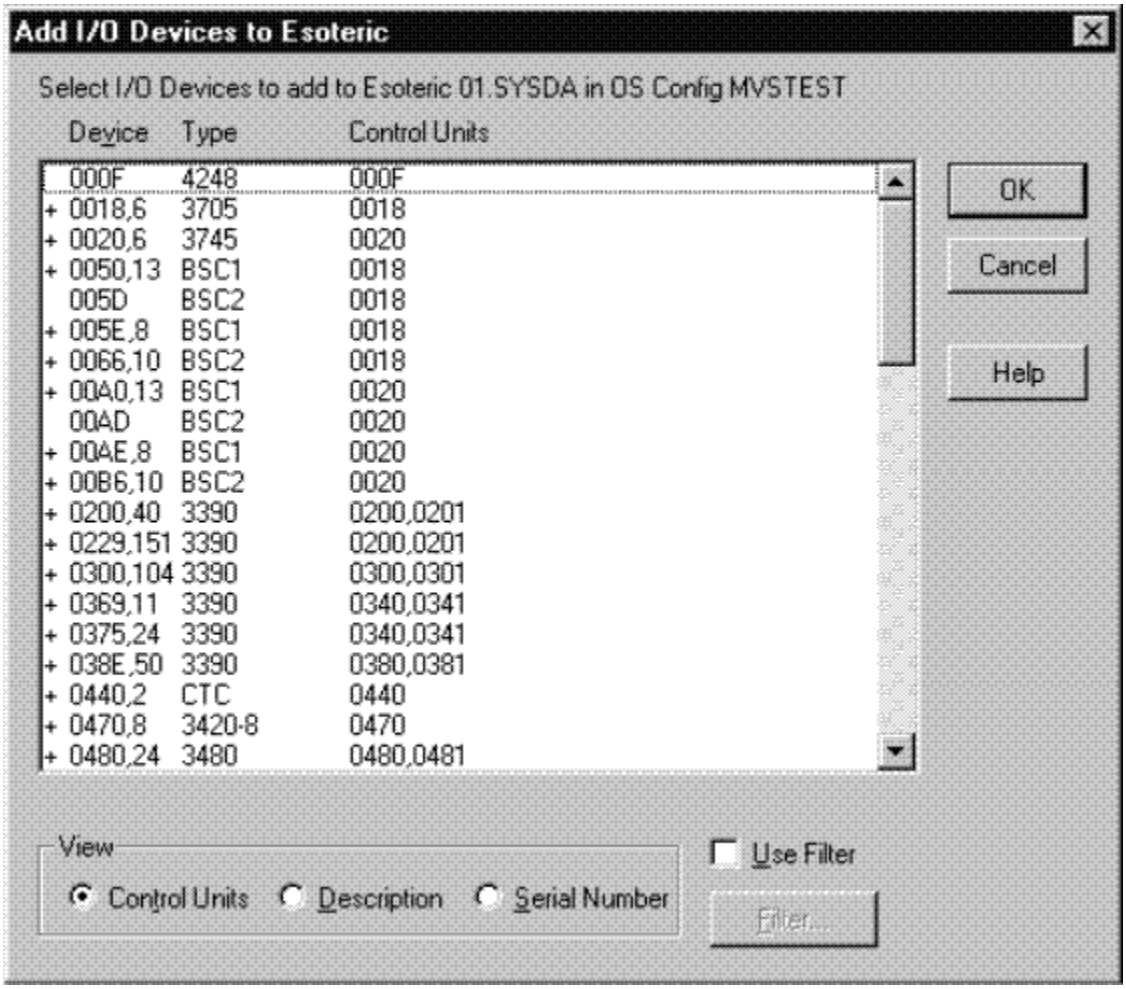


Figure 171. Add I/O Devices to Esoteric dialog

Removing devices from an esoteric

Select one or more devices and press the **Remove** button in the **List Devices for Esoteric** dialog. HCM displays 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 z/OS 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).

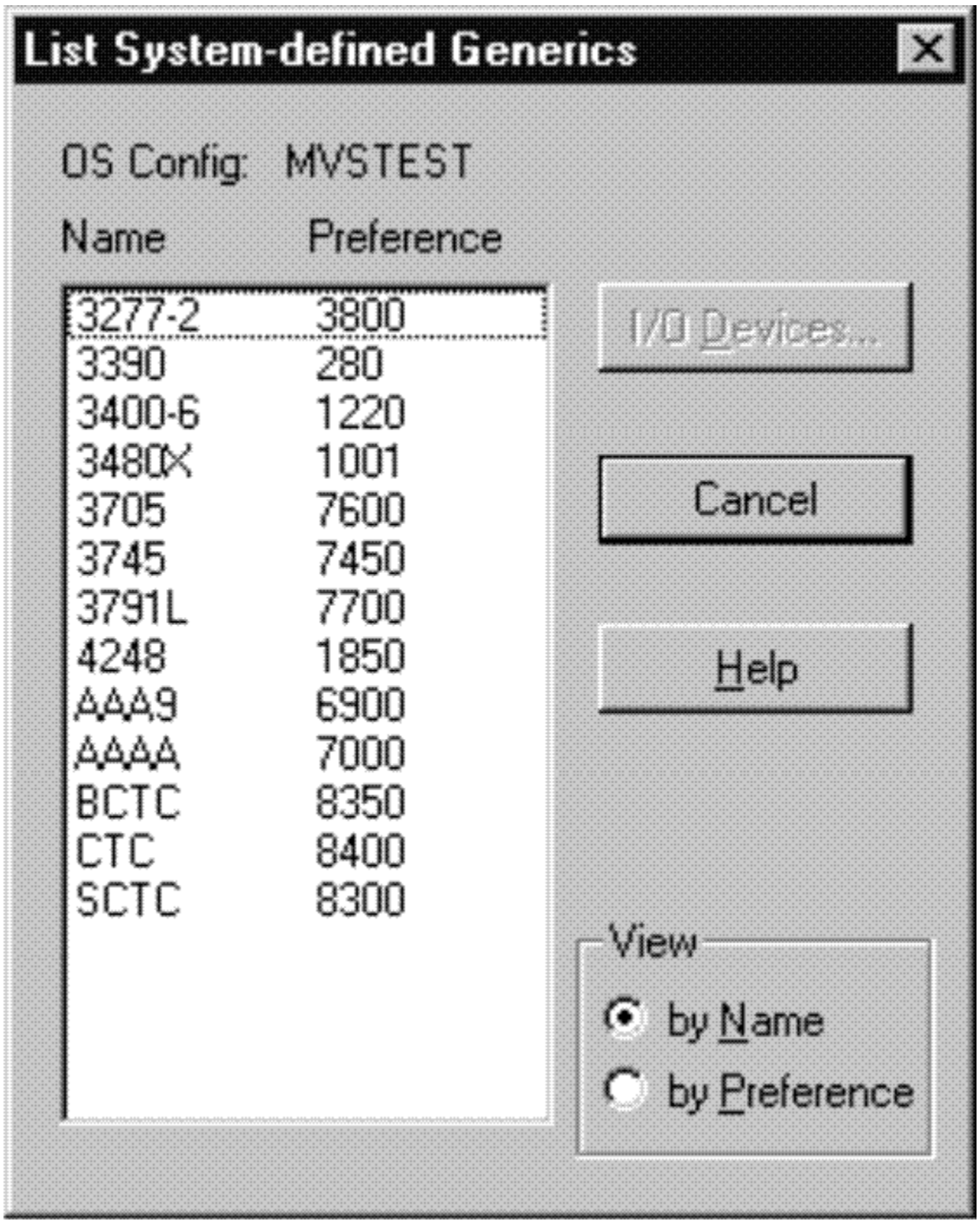


Figure 172. List System-Defined Generics dialog

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).

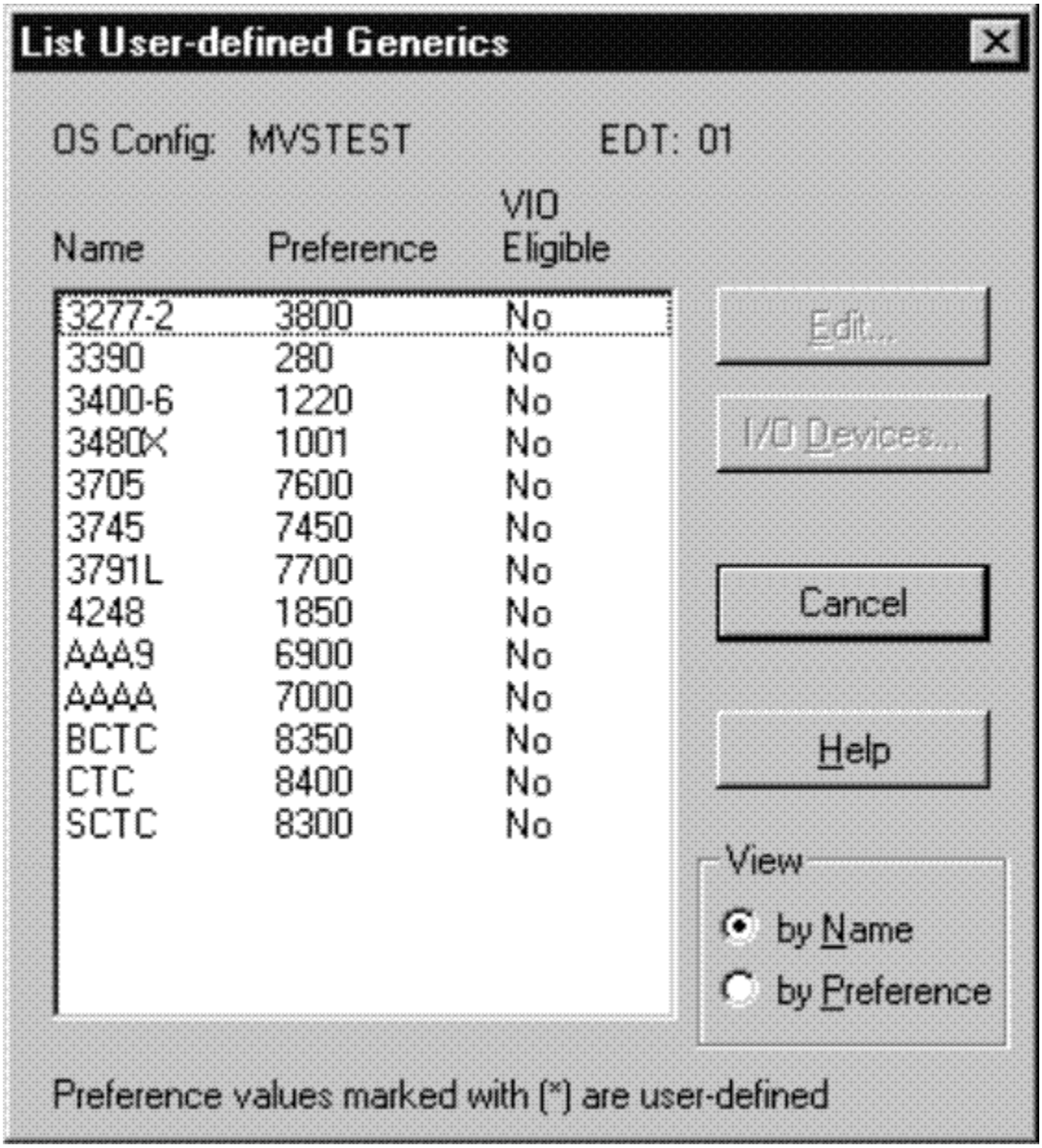


Figure 173. List User-Defined Generic dialog

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. 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) affects the appearance of the configuration diagram on the screen.

Besides printing the generated text reports, you can store them in softcopy formats:

- the ***XML/HTML*** file format enables users, for example, to view the generated XML reports as HTML files using a web browser.
- the ***CSV*** file format for example, allows you to load HCM reports into spreadsheet programs for further processing.

Tips:

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.

Before printing diagram reports, turn off the display of cabinets using the **View** menu. This will reduce the number of pages per diagram.

Printing configuration diagrams

You can print configuration diagrams with the **Print Diagram...** command under the **File** menu. First ensure that 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. You can either specify a certain scaling factor or you can have HCM calculate the optimal scaling factor which ensures that all texts are readable. Row/column numbers are printed on each page to allow a correct placing when combining the single pages to a complete picture.

Tip:

Screen snapshots typically look best if printed in landscape mode.

© Copyright IBM Corp. 1995, 2021

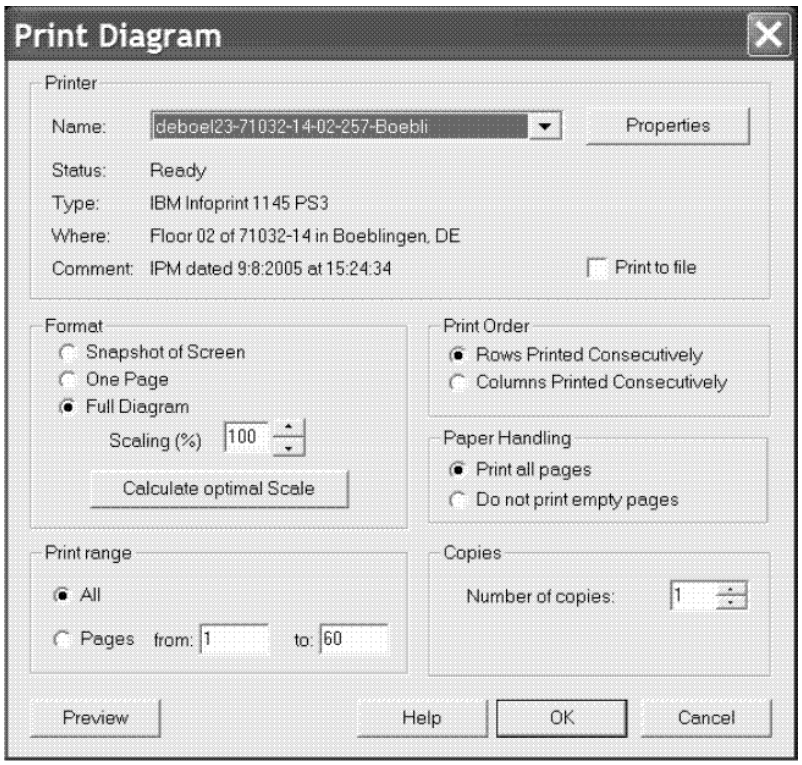


Figure 174. Printing a configuration diagram

There are the following 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:**. 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 as indicated in the **Page range** section and at the scale specified in the **Scaling (%)** field. You can select any scaling factor between 20% and 100%. Use the **Preview** button to see how the diagram will be split between pages.
- **Calculate optimal Scale**: Prints the complete diagram as small as possible, but so that all contained text is still readable. Pressing this button selects the **Full Diagram** radio button and calculates the optimal scale and displays this value in the editable **Scaling (%)** field. Note that in most cases still several pages will be required to print the scaled diagram.

Generating text reports and diagram reports

In HCM, you can generate and print text reports and diagram reports. A *diagram report* is a configuration diagram showing one particular object of interest and the equipment connected to it.

You can select from the following output formats:

- ***XML/HTML*** saves all selected reports in an XML file format to a user-specified directory. HCM provides the appropriate XSL and CSS style sheets and XSD schema definitions to enable users, for example, to view the generated XML reports as HTML files using a web browser.
- ***CSV*** saves all selected reports in an CSV (comma separated values) file format to a user-specified directory. The CSV file format, for example, allows you to load HCM reports into spreadsheet programs for further processing.

  The applications which you want to use for further processing the generated CSV files, may expect different separation characters. Therefore, if you select the *CSV* option, HCM offers a dialog to set the appropriate separation character before it generates the output.

- ***Printer*** prints all selected reports on the current printer, or, depending on your **Page Setup** selection, stores the reports in your specified file.

In stand-alone mode, you can print complete reports only from a production configuration file (see "Printing reports, diagrams, cable labels" on page 96).

Before printing a text report or a diagram report, you can specify parameters for the page layout and printer options. Open the **Page Setup** dialog by choosing **Page Setup...** from the **File** menu. Here you can select, for example, the orientation, (**Portrait** is recommended), a scaling factor and a printer.

Now open the **Generate Reports** dialog by choosing **Generate HCM Reports...** from the **File** menu. This dialog allows you to set options for the reports you are generating:

- **Output Format**: Here you specify the format of the generated report (one of XML/HTML, CSV or Printer).
- **Report Description**: The **Title Page and Contents** and the **Index** check box are only selectable for the **Printer** output format. **Report Title** and **Note 1** and **Note 2** are available for any output format.

  Checking the **Title Page and Contents** box, you can generate a title page and a table of contents for the report. The table of contents lists the sections that you include and the pages where each begins. This option is selected by default, but you may want to suppress these pages when printing reports on individual objects.

  The title page optionally contains the report title and two notes. If for your current configuration, you have specified a title and footnotes (using the **Titles & Footnotes...** dialog from the **Edit** menu), then these texts appear as a default in the **Report Title** and **Note 1, Note 2** fields. You may overtype any of these for your current report generation.

  Selecting the **Index** check box, generates an index in the report output. This choice also enables the **Index Options...** button. Pressing this button invokes the ***Index Options*** dialog which lets you select which user fields should be included as optional index entries in the report. For example, clicking on *User Field 01* 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 (see also "User data fields" on page 29).
- **Scope**: Lets you specify whether you want to generate a report for all objects, visible objects only, selected objects only, or for certain objects only, that you specify in the **Limitations** section.
- **Text Reports**: Lets you specify the objects and further details of the specified objects, for which you want to generate a detailed report depending on the scope:
  - **Processor**: *Processor Summary* prints for each processor a summary that includes a description, and lists of partitions and CHPID types. *Channel Path Connections* 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.
  - **Cables and Links**: *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. *Used Cables* prints all cables that are connected to the set of objects specified by All, Visible or Selected options. *Unused Cables* prints all cables that are connected to the set of objects specified by All, Visible, Selected or Limited options.
  - **Cabinet and General Box**: creates a tabular view for cabinets (specified by All, Visible or Selected) showing all connections to each patchport. You may generate a separate report for the *Front View* and *Back View* of each panel showing its patchports arrayed as they appear on the panel when viewed from the appropriate side.
  - **Switches, Controllers, Strings, Crossbar Switches**: specifies the types of objects for which detailed text reports and diagram reports will be printed according to your selected scope.
- **Diagram Reports**: This section is only usable for the **Printer** output format. It lets you specify the objects, for which a diagram report (depending on the scope) should be generated. Note that you cannot print a diagram report for cable or cabinet objects. These checkboxes are only available for output format Printer and not available for scope **Limited**.
- **Limitations**: If you selected the **Limited** radio button in the **Scope** section, you can further limit the report to certain (single or multiple) objects for each object category.

Using the ***Save Options...*** button, you can save a certain set of report options into an option file. At a later point in time, using the ***Load Options...*** button, you can load a set of report options from the related option file. HCM invokes the **Save/Load Report Option File** dialogs respectively. This functionality helps you to quickly generate reports that you need to obtain frequently.

Note: Option files use the Windows *.ini* file format.

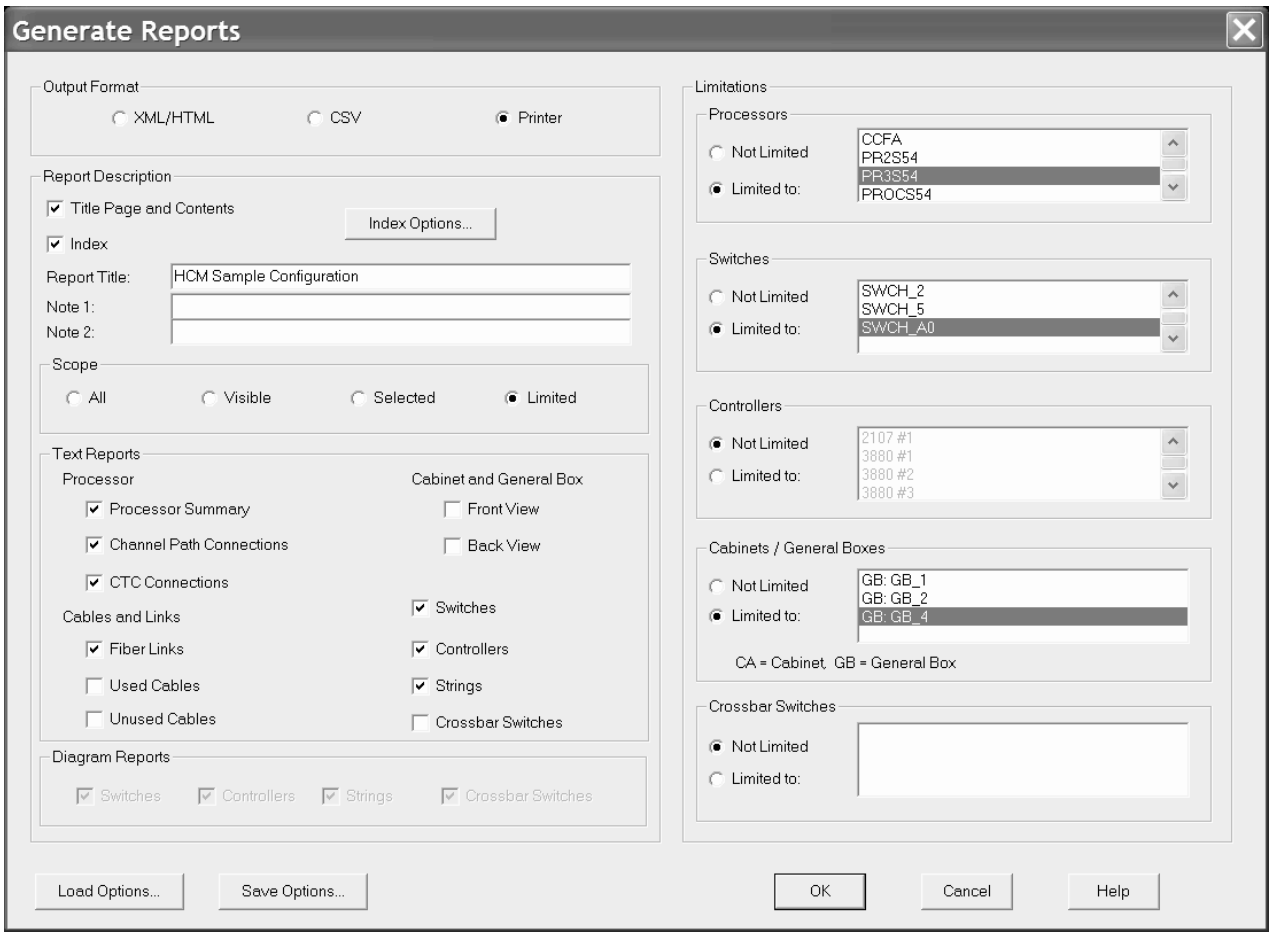


Figure 175. Generate Reports

Tip 1:

To print a diagram report 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 diagram report using the One Page preview or a Snapshot of Screen options in the **Print Diagram** dialog.

Tip 2:

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 text reports and diagram reports

Explore the HCM report generation facilities (to be invoked by selecting **Generate HCM Reports...** from the **File** menu). To receive an impression how the same report would look when produced in the various available output formats, have a look at Figure 176 on page 267 through Figure 178 on page 269: These

figures show a part of a ***Processor Summary*** report as printed on paper, displayed by a web browser and processed with a spreadsheet program.

Hereafter, "Printed sample reports" on page 270 presents examples of some of the types of text reports and diagrams generated to be printed and "Sample cable labels" on page 276 presents a printout example of a cable labels.

K25 2096-S07 "z9 BC with basic support" [25]
Mode: LPAR
Support level: XMP, 2096-S07 support
Network name: IBM390PS
CPC name: K25
Local system name:

Channel subsystem K25.0
Partitions (by image number)
1 JG0 1 OS "43 JG0 on P24"

2 ZOSEJH0 2 OS "44 JH0 PUMA image on P24"

CHPID	Type	Managed	I/O Cluster	# of Devices	MFS	Mode	PCHID	> 160 TCP/IP stacks	Partition Numbers 1	2
00	CBP			21	0	SHR	018		A	A
01	CBP			28	0	SHR	01E		A	A
04	CBP			21	0	SHR	2B0		A	A
05	CNC	Yes	UTCPLXJ8	0	0	SHR	160			
09	CNC			897	0	SHR	170		A	A
0A	CNC			384	0	SHR	1E8		A	A
0F	CNC			384	0	SHR	1E0		A	A
10	CNC			0	0	SHR	1F0		A	A
12	CNC			384	0	SHR	161		A	A
13	CFP			28	0	SHR	190		A	A
14	CFP			28	0	SHR	118		A	A
15	CFP			28	0	SHR	181		A	A
16	CFP			0	0	SHR	191		A	A
17	CFP			21	0	SHR	189		A	A
18	CFP			21	0	SHR	119		A	A
1F	CNC			384	0	SHR	171		A	A
20	OSD			16	0	SHR	1B0	No	A	A
21	OSC			16	0	SHR	1B1		A	A
22	OSD			16	0	SHR	201	No	A	A
23	OSD			16	0	SHR	1C0	No	A	A
30	CNC	Yes	UTCPLXJ8	0	0	SHR	266			
31	CNC	Yes	UTCPLXJ8	0	0	SHR	1F6			
32	CNC	Yes	UTCPLXJ8	0	0	SHR	178			
33	CNC	Yes	UTCPLXJ8	0	0	SHR	1E6			
3B	CNC			384	0	SHR	1F1		A	A
48	CNC			384	0	SHR	261		A	A
50	CNC			384	0	SHR	162		A	A
63	CNC			384	0	SHR	172		A	A
7F	CNC			8	0	SHR	1EA		A	A
84	CFP			21	0	SHR	180		A	A
85	CFP			21	0	SHR	188		A	A
88	CNC			32	0	SHR	1E2		A	A
8B	CNC			552	0	SHR	262		A	A
90	CNC			512	0	SHR	163		A	A
91	CNC			84	0	SHR	173		A	A
92	CNC			171	0	SHR	1EB		A	A
98	CNC	Yes	UTCPLXJ8	0	0	SHR	1E3			
99	CNC			325	0	SHR	1F3		A	A
9B	CNC			513	0	SHR	263		A	A
A3	CNC			256	0	SHR	164		A	A
B0	CNC			85	0	SHR	174		A	A
B1	CNC			288	0	SHR	1EC		A	A
B2	CNC			257	0	SHR	1E4		A	A
B3	CTC			81	0	SHR	1F4		A	A
B4	CTC			0	0	SHR	1E5		A	A
BF	CVC			0	0	DED	267			A
C3	CNC	Yes	UTCPLXJ8	0	0	SHR	165			
D3	CNC			512	0	SHR	177		A	A

Figure 176. Printer format sample

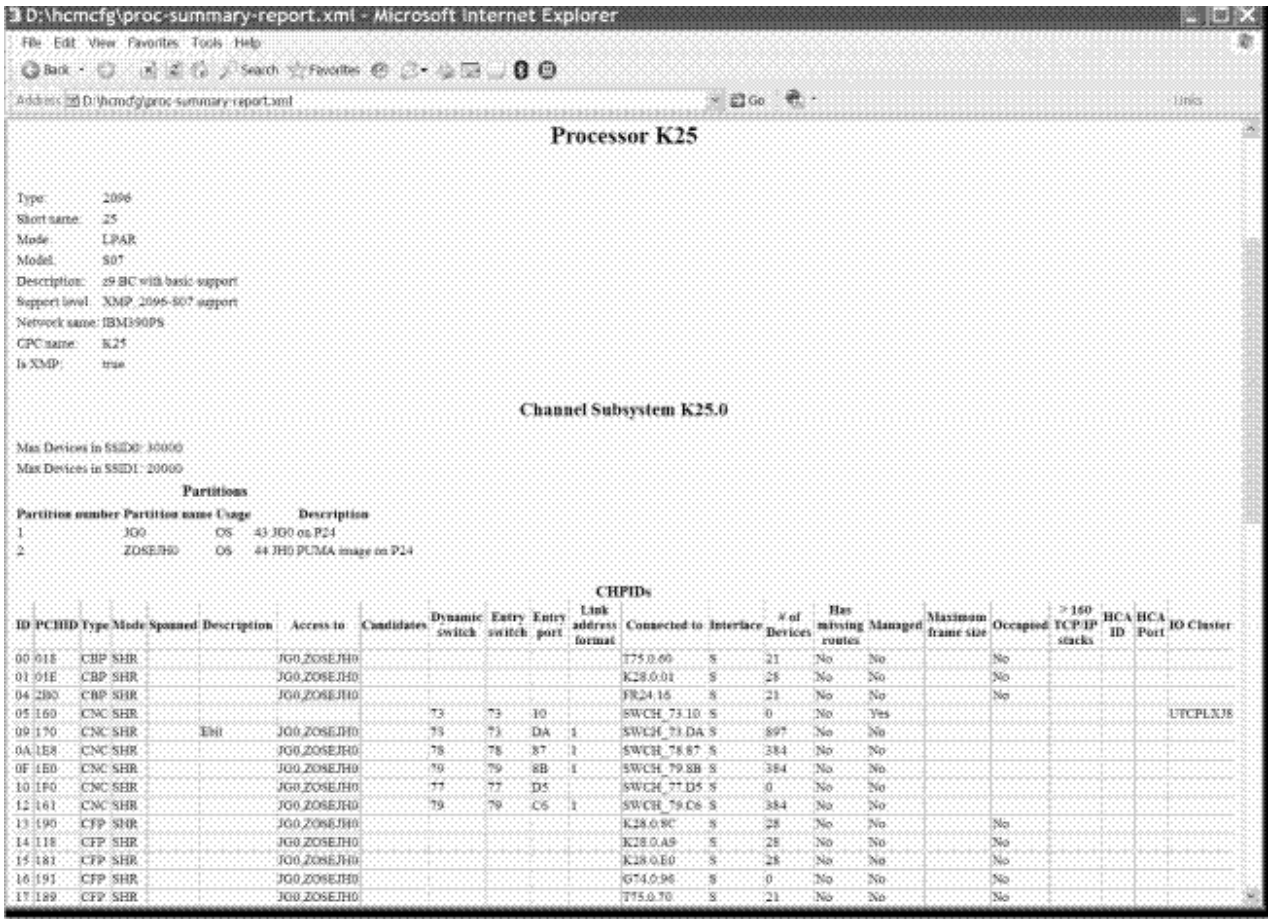
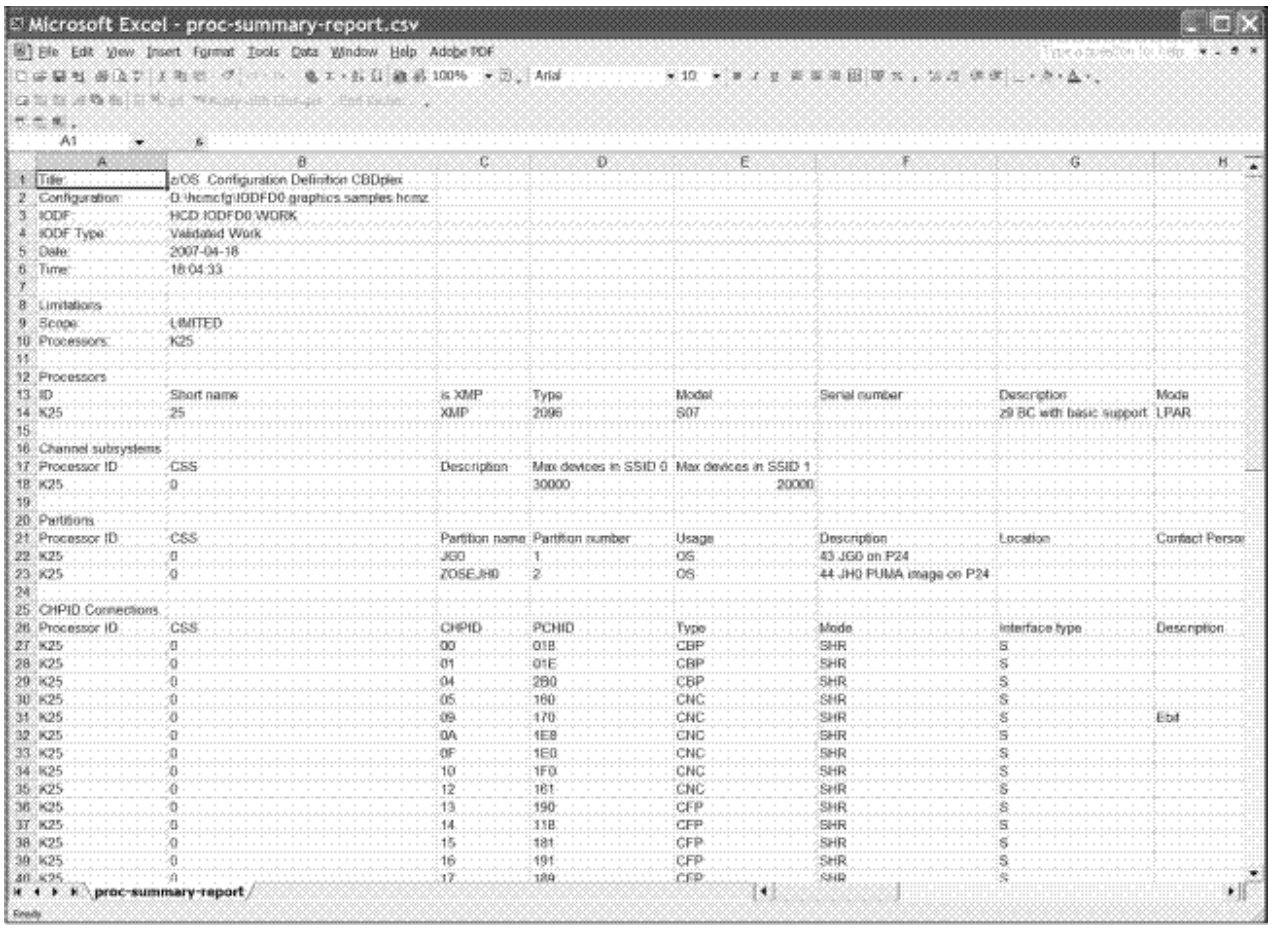


Figure 177. XML/HTML format sample


Microsoft Excel - proc-summary-report.csv

File Edit View Insert Format Tools Data Window Help Adobe PDF

	A	B	C	D	E	F	G	H
1	Title:	z/OS Configuration Definition CBDplex						
2	Configuration:	D:\hcmcfg\IODFD0.graphics.samples.hcmz						
3	IODF:	HCD.IODFD0.WORK						
4	IODF Type:	Validated Work						
5	Date:	2007-04-18						
6	Time:	18:04:33						
7								
8	Limitations:							
9	Scope:	LIMITED						
10	Processors:	K25						
11								
12	Processors							
13	ID	Short name	is XMP	Type	Model	Serial number	Description	Mode
14	K25	25	XMP	2096	S07		z9 BC with basic support	LPAR
15								
16	Channel subsystems							
17	Processor ID	CSS	Description	Max devices in SSID 0	Max devices in SSID 1			
18	K25	0		30000	20000			
19								
20	Partitions							
21	Processor ID	CSS	Partition name	Partition number	Usage	Description	Location	Contact Perso
22	K25	0	JG0	1	OS	43 JG0 on P24		
23	K25	0	ZOSEJH0	2	OS	44 JH0 PUMA image on P24		
24								
25	CHPID Connections							
26	Processor ID	CSS	CHPID	PCHID	Type	Mode	Interface type	Description
27	K25	0	00	018	CBP	SHR	S	
28	K25	0	01	01E	CBP	SHR	S	
29	K25	0	04	2B0	CBP	SHR	S	
30	K25	0	05	160	CNC	SHR	S	
31	K25	0	09	170	CNC	SHR	S	Ebit
32	K25	0	0A	1E8	CNC	SHR	S	
33	K25	0	0F	1E0	CNC	SHR	S	
34	K25	0	10	1F0	CNC	SHR	S	
35	K25	0	12	161	CNC	SHR	S	
36	K25	0	13	190	CFP	SHR	S	
37	K25	0	14	118	CFP	SHR	S	
38	K25	0	15	181	CFP	SHR	S	
39	K25	0	16	191	CFP	SHR	S	
40	K25	0	17	189	CFP	SHR	S	

proc-summary-report

Ready

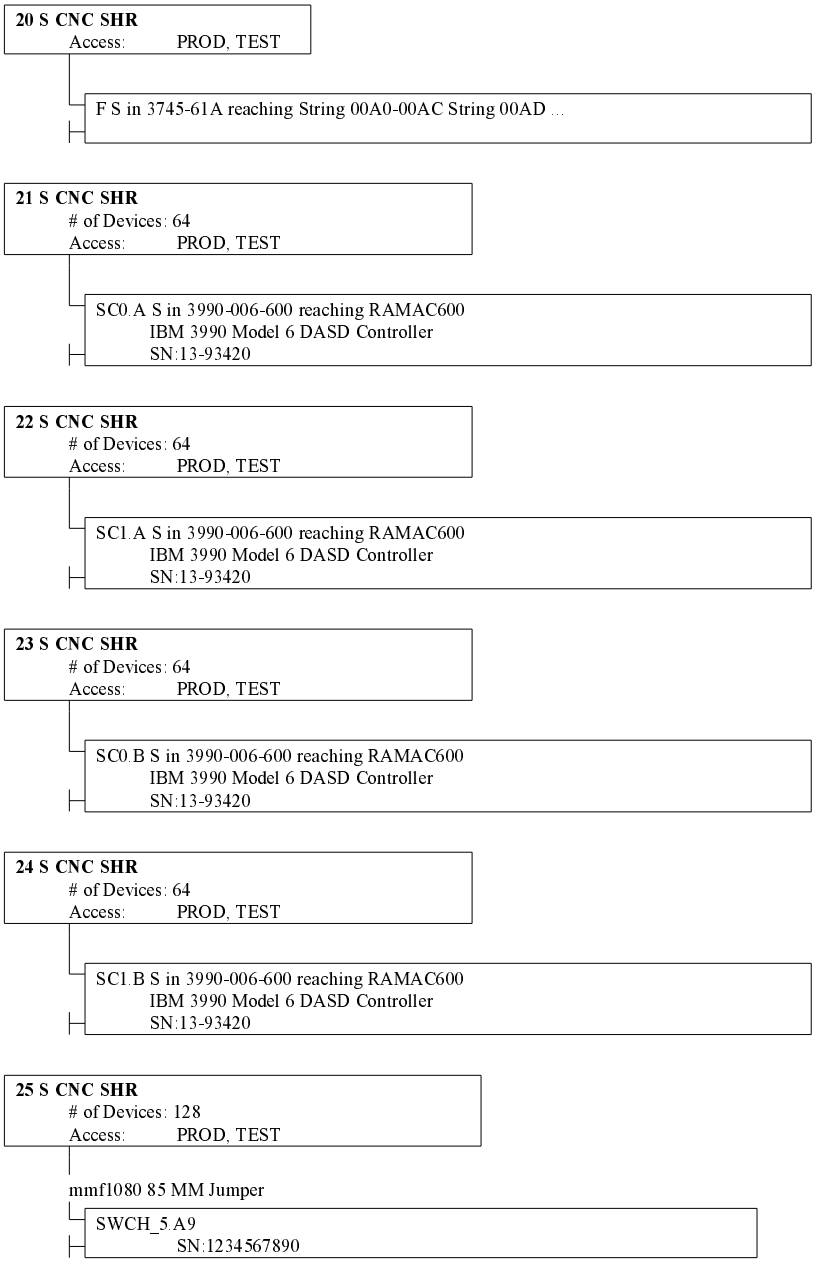
Figure 178. CSV format sample

Printed sample reports

Channel Path Connections

BOESYS 9672-R41 "OS/390 Development System"
SN:0000009672

Asset Mgmt .#: 8337539-77835
Location: Main Complex



Hardware Configuration Manager 2001-10-11 14:46 1

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 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 ==**
3990-003-200.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 ==**

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 -mmf1025- C10A2 -trunk- C20A2 -mmf1070- 3990-003-240.SC0.B
n/a	n/a	SWCH_2.85 -mmf1029- C10A6 -trunk- C20A6 -mmf1066- 3990-003-200.SC0.B
n/a	n/a	SWCH_2.86 -mmf1033- C10B4 -trunk- C20B4 -mmf1062- 3990-006-280.SC0.F
n/a	n/a	SWCH_2.87 -mmf1037- C10C2 -trunk- C20C2 -mmf1058- 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 -mmf1027- C10A4 -trunk- C20A4 -mmf1072- 3990-003-200.SC1.B
n/a	n/a	SWCH_2.8D -mmf1031- C10B2 -trunk- C20B2 -mmf1068- 3990-003-240.SC1.B
n/a	n/a	SWCH_2.8E -mmf1035- C10B6 -trunk- C20B6 -mmf1064- 3990-006-280.SC1.F
n/a	n/a	SWCH_2.8F -mmf1039- C10C4 -trunk- C20C4 -mmf1060- 3990-006-300.SC1.F
n/a	n/a	SWCH_2.94 -mmf1045- C10C6 -jumper- 3990-003-360.SC0.F
n/a	n/a	SWCH_2.95 -mmf1041- C10D4 -trunk- C20D4 -mmf1049- 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 -mmf1043- C10D6 -trunk- C20D6 -mmf1051- 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 -mmf1024- C10A1 -trunk- C20A1 -mmf1069- 3990-003-240.SC0.A
n/a	n/a	SWCH_5.81 -mmf1028- C10A5 -trunk- C20A5 -mmf1065- 3990-003-200.SC0.A
n/a	n/a	SWCH_5.82 -mmf1032- C10B3 -trunk- C20B3 -mmf1061- 3990-006-280.SC0.E
n/a	n/a	SWCH_5.83 -mmf1036- C10C1 -trunk- C20C1 -mmf1057- 3990-006-300.SC0.E
n/a	n/a	SWCH_5.84 -jumper- S->P
n/a	n/a	SWCH_5.88 -mmf1026- C10A3 -trunk- C20A3 -mmf1071- 3990-003-200.SC1.A
n/a	n/a	SWCH_5.89 -mmf1030- C10B1 -trunk- C20B1 -mmf1067- 3990-003-240.SC1.A
n/a	n/a	SWCH_5.8A -mmf1034- C10B5 -trunk- C20B5 -mmf1063- 3990-006-280.SC1.E
n/a	n/a	SWCH_5.8B -mmf1038- C10C3 -trunk- C20C3 -mmf1059- 3990-006-300.SC1.E
n/a	n/a	SWCH_5.90 -mmf1044- C10C5 -trunk- C20C5 -mmf1052- 3990-003-360.SC0.E
n/a	n/a	SWCH_5.91 -mmf1040- C10D3 -trunk- C20D3 -mmf1048- 3990-006-380.SC0.E
n/a	n/a	SWCH_5.98 -mmf1046- C10D1 -jumper- 3990-003-360.SC1.E
n/a	n/a	SWCH_5.99 -mmf1042- C10D5 -trunk- C20D5 -mmf1050- 3990-006-380.SC1.E
n/a	n/a	SWCH_5.A2 -mmf1078- BOESYS.26
n/a	n/a	SWCH_5.A6 -mmf1079- BOESYS.27
n/a	n/a	SWCH_5.A9 -mmf1080- BOESYS.25
n/a	n/a	SWCH_5.AA -mmf1081- 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

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

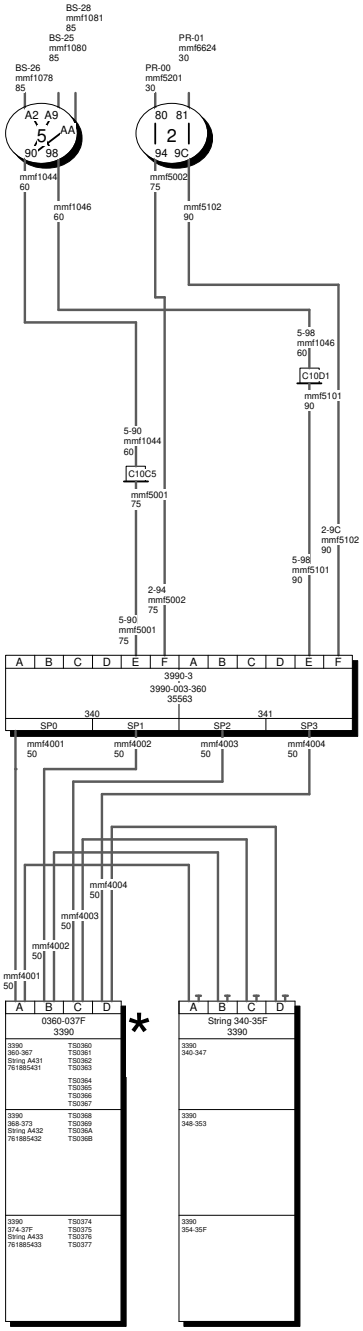
String A432
SN:761885432
Asset Mgmt .#: 8337539-77835
Location: Main Complex
Usage: Databases

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

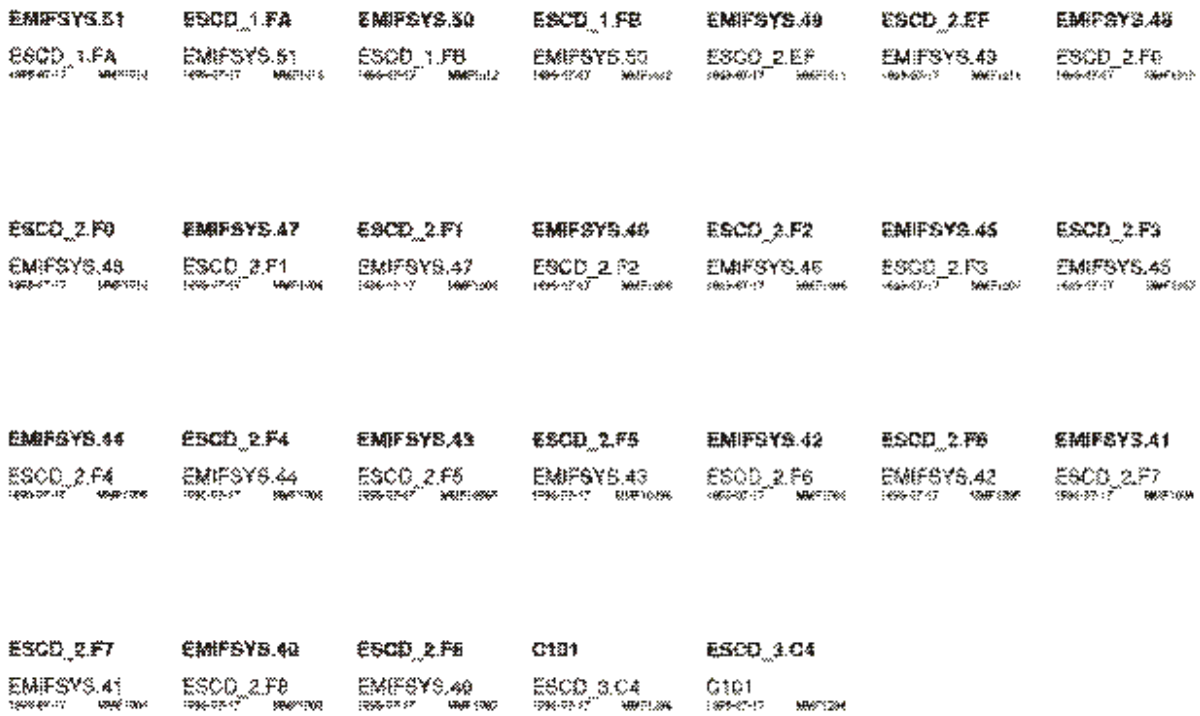


Cables

From	ID	Len	Type	Page #	To	db Loss A & B	
ESCD_5.80	mmf1024	60	MM Jumper		C10A1	n/a	n/a
ESCD_5.88	mmf1026	60	MM Jumper		C10A3	n/a	n/a
ESCD_5.81	mmf1028	60	MM Jumper		C10A5	n/a	n/a
ESCD_5.89	mmf1030	60	MM Jumper		C10B1	n/a	n/a
ESCD_5.82	mmf1032	60	MM Jumper		C10B3	n/a	n/a
ESCD_5.8A	mmf1034	60	MM Jumper		C10B5	n/a	n/a
ESCD_5.83	mmf1036	60	MM Jumper		C10C1	n/a	n/a
ESCD_5.8B	mmf1038	60	MM Jumper		C10C3	n/a	n/a
ESCD_5.91	mmf1040	60	MM Jumper		C10D3	n/a	n/a
ESCD_5.99	mmf1042	60	MM Jumper		C10D5	n/a	n/a
ESCD_5.90	mmf1044	60	MM Jumper		C10C5	n/a	n/a
ESCD_5.98	mmf1046	60	MM Jumper		C10D1	n/a	n/a
C20D3	mmf1048	100	MM Jumper		3990-006-380.SC0.E	n/a	n/a
C20D5	mmf1050	100	MM Jumper		3990-006-380.SC1.E	n/a	n/a
C20C1	mmf1057	100	MM Jumper		3990-006-300.SC0.E	n/a	n/a
C20C3	mmf1059	100	MM Jumper		3990-006-300.SC1.E	n/a	n/a
C20B3	mmf1061	100	MM Jumper		3990-006-280.SC0.E	n/a	n/a
C20B5	mmf1063	100	MM Jumper		3990-006-280.SC1.E	n/a	n/a
C20A5	mmf1065	100	MM Jumper		3990-003-200.SC0.A	n/a	n/a
C20B1	mmf1067	100	MM Jumper		3990-003-240.SC1.A	n/a	n/a
C20A1	mmf1069	100	MM Jumper		3990-003-240.SC0.A	n/a	n/a
C20A3	mmf1071	100	MM Jumper		3990-003-200.SC1.A	n/a	n/a
BOESYS.26	mmf1078	85	MM Jumper		ESCD_5.A2	n/a	n/a
BOESYS.27	mmf1079	85	MM Jumper		ESCD_5.A6	n/a	n/a
BOESYS.25	mmf1080	85	MM Jumper		ESCD_5.A9	n/a	n/a
BOESYS.28	mmf1081	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 standard printable label form. To receive such a printout, select **Print Cable Labels...** from the **File** menu.



The following is a sample of cable labels as saved in a comma delimited ASCII file.

```
SWCH_F1.10,1750 #1.01.A0[00],2006-02-28,????,SWCH_F1.10,1750 #1.01.A0[00]
1750 #1.01.A0[00],SWCH_F1.10,2006-02-28,????,SWCH_F1.10,1750 #1.01.A0[00]
SWCH_F2.10,1750 #1.02.A1[04],2006-02-28,????,SWCH_F2.10,1750 #1.02.A1[04]
1750 #1.02.A1[04],SWCH_F2.10,2006-02-28,????,SWCH_F2.10,1750 #1.02.A1[04]
```

```
1750 #2.C1,1750 #1.C1,2006-02-28,????,1750 #2.C1,1750 #1.C1
1750 #1.C1,1750 #2.C1,2006-02-28,????,1750 #2.C1,1750 #1.C1
ERV01.0.F2,ERV01.0.F7,2006-02-28,CF03,ERV01.0.F2,ERV01.0.F7
ERV01.0.F7,ERV01.0.F2,2006-02-28,CF03,ERV01.0.F2,ERV01.0.F7
ERV01.0.F1,ERV01.0.F5,2006-02-28,CF02,ERV01.0.F1,ERV01.0.F5
ERV01.0.F5,ERV01.0.F1,2006-02-28,CF02,ERV01.0.F1,ERV01.0.F5
ERV01.0.F3,ERV01.0.F6,2006-02-28,CF04,ERV01.0.F3,ERV01.0.F6
ERV01.0.F6,ERV01.0.F3,2006-02-28,CF04,ERV01.0.F3,ERV01.0.F6
ERV01.2.E0(1E0),ERV02.1.E0(1E0),2006-02-28,CF05,ERV01.2.E0(1E0),ERV02.1.E0(1E0)
ERV02.1.E0(1E0),ERV01.2.E0(1E0),2006-02-28,CF05,ERV01.2.E0(1E0),ERV02.1.E0(1E0)
ERV01.2.E1(1E1),ERV02.1.E1(1E1),2006-02-28,CF06,ERV01.2.E1(1E1),ERV02.1.E1(1E1)
ERV02.1.E1(1E1),ERV01.2.E1(1E1),2006-02-28,CF06,ERV01.2.E1(1E1),ERV02.1.E1(1E1)
String 0060-007F.B,3174 #2.A,2006-02-28,????,3174 #2.A,String 0060-007F.B
String 0060-007F.B,String 0080-009F.B,2006-02-28,????,3174 #2.A,String 0080-009F.B
String 0080-009F.B,String 0060-007F.B,2006-02-28,????,3174 #2.A,String 0080-009F.B
ERV03A2.1.01(101),ERV03A2.1.05(105),2006-02-28,A026,ERV03A2.1.01(101),ERV03A2.1.05(105)
ERV03A2.1.05(105),ERV03A2.1.01(101),2006-02-28,A026,ERV03A2.1.01(101),ERV03A2.1.05(105)
ERV03A2.3.00(200),ERV03A2.3.04(204),2006-02-28,A030,ERV03A2.3.00(200),ERV03A2.3.04(204)
ERV03A2.3.04(204),ERV03A2.3.00(200),2006-02-28,A030,ERV03A2.3.00(200),ERV03A2.3.04(204)
ERV03A2.3.01(201),ERV03A2.3.05(205),2006-02-28,A031,ERV03A2.3.01(201),ERV03A2.3.05(205)
ERV03A2.3.05(205),ERV03A2.3.01(201),2006-02-28,A031,ERV03A2.3.01(201),ERV03A2.3.05(205)
   ↑                ↑                ↑        ↑      ↑                ↑
   |                |                |        |      |                |
   |                |                |        |      |                |
   |                |                |        |      |                |_ Link target
   |                |                |        |      |                   (with PCHID,
   |                |                |        |      |                    if possible)
   |                |                |        |      |_____ Link source
   |                |                |        |             (with PCHID, if possible)
   |                |                |        |
   |                |                |        |_____________  Cable ID
   |                |                |                        (????, if no ID provided)
   |                |                |
   |                |                |_________________________ today's date
   |                |
   |                |_________ Cable destination (with PCHID, if possible)
   |
   |____________ Cable source (with PCHID, if possible)
```

Viewing HCD configuration reports

Besides the reports produced by HCM, you can also view HCD configuration reports on your HCM workstation in connected mode. Use the pull-down choice **View IODF Reports...** from the **File** menu to invoke the **IODF Reports** dialog (Figure 179 on page 279) from where you can select HCD configuration reports about the logical I/O configuration data in the current IODF. Optionally, you can limit the reports to selected objects (for example, a processor or an OS configuration).

Also, for these reports, you can select from various output formats. You can view the contents of the selected reports in a window and optionally you can save the output into a file on your workstation. Or you can save the reports in CSV (comma separated values) format for later processing the data, and you can also save the reports in XML format. To view the generated XML reports in a web browser, you need to provide the appropriate XSL and CSL style sheets and XSD schema definitions.

Note: The applications which you want to use for further processing the generated CSV files, may expect different separation characters. Therefore, if you select the *CSV* option, HCM offers a dialog to set the appropriate separation character before it generates the output.

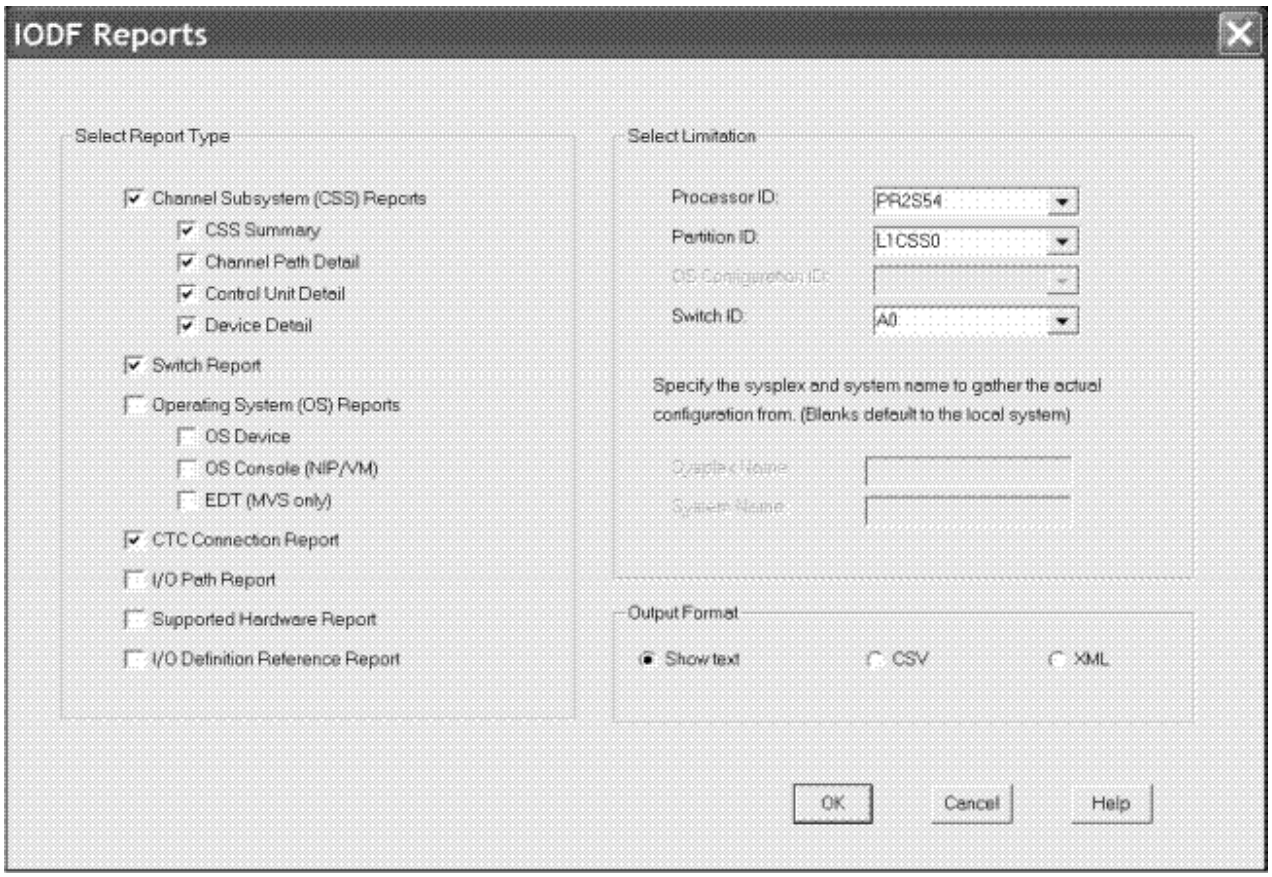


Figure 179. IODF Reports

Viewing HCD compare reports

On your HCM workstation you can view IODF compare reports produced by HCD on the host. These reports are displayed in an HCM window, and optionally you can save the output from this window to a file.

There are two flavors of this compare facility. You can:

- view reports that compare two different IODFs (see "Viewing IODF compare reports" on page 279)
- view reports that compare the hardware (channel subsystem/CSS) and software (operating system/OS) definition in the currently accessed IODF. Such reports specifically compare the device definitions of a selected CSS and OS configuration, showing which devices (their numbers and types) are defined to either the CSS or the OS, or both (see "Viewing CSS/OS compare reports" on page 280).

For information about the contents and layout of these reports, refer to the *z/OS HCD User's Guide*.

Viewing IODF compare reports

You can view HCD IODF compare reports on your workstation. Select **View IODF Compare Reports...** from the **File** menu bar choice. The following dialog appears, in which you can specify report types and limitations:

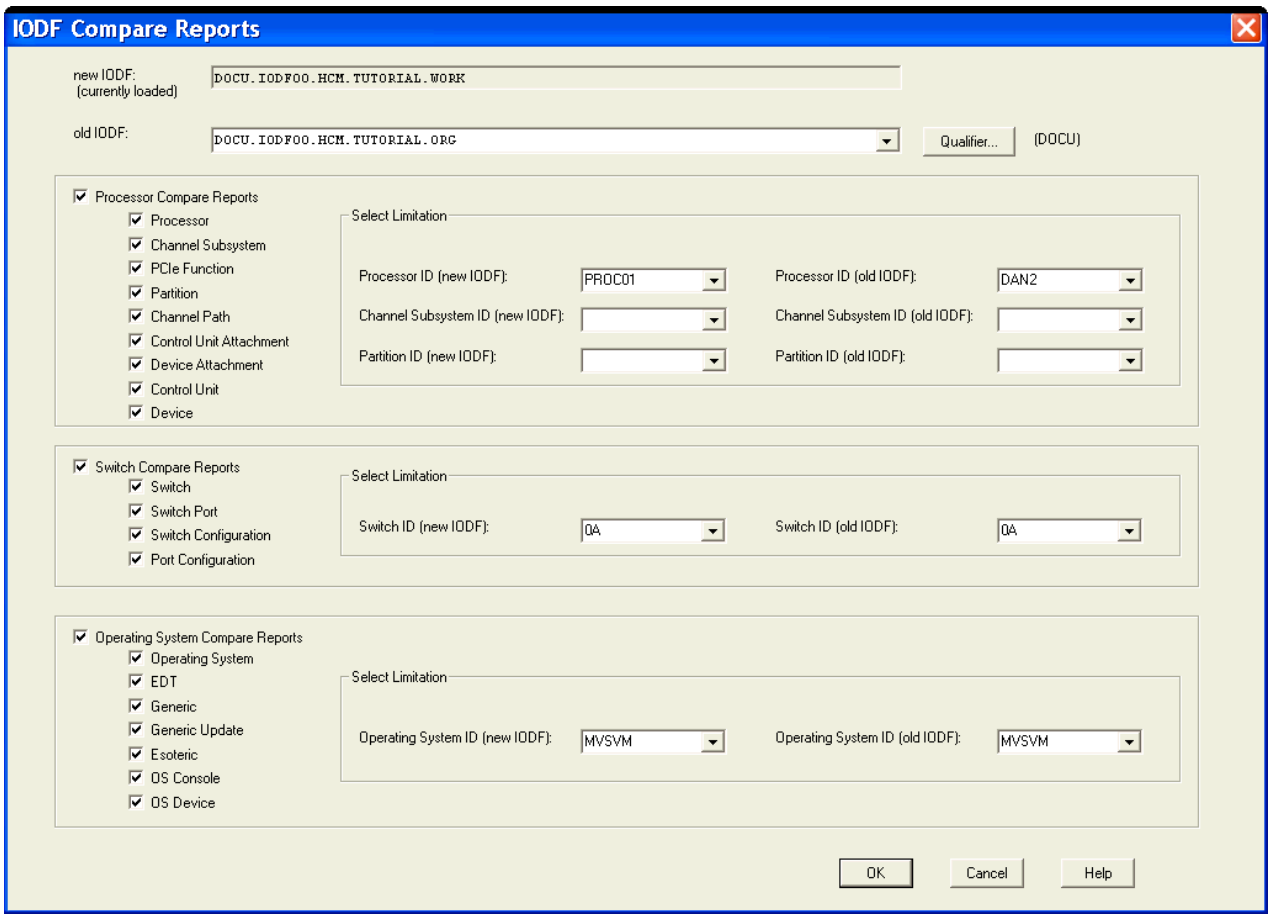


Figure 180. IODF Compare Reports

Specify your selections and press the **OK** button. HCM invokes another dialog that lets you specify one or more options of what information you want to see:

- ***Print inserted data*** to show added data of definitions, connections, or attachments from the new IODF, or of added attributes or relations of existing items.
- ***Print deleted data*** to show deleted data of definitions, connections, or attachments from the old IODF, or of deleted attributes or relations of existing items.
- ***Print unchanged data*** to show data of unchanged definitions, connections, or attachments, as well as their unchanged attributes or relations.
- ***Print unchanged item IDs*** to show IDs of all definitions, connections, or attachments whose values have not been changed.

Viewing CSS/OS compare reports

You can view HCD CSS/OS compare reports on your workstation. Use this facility to display and save reports that compare devices of an CSS or OS within one IODF.

You can invoke this function selecting **View IODF Compare CSS/OS Report...** from the **File** menu bar choice. The following dialog appears, in which you specify your CSS and OS limitations:

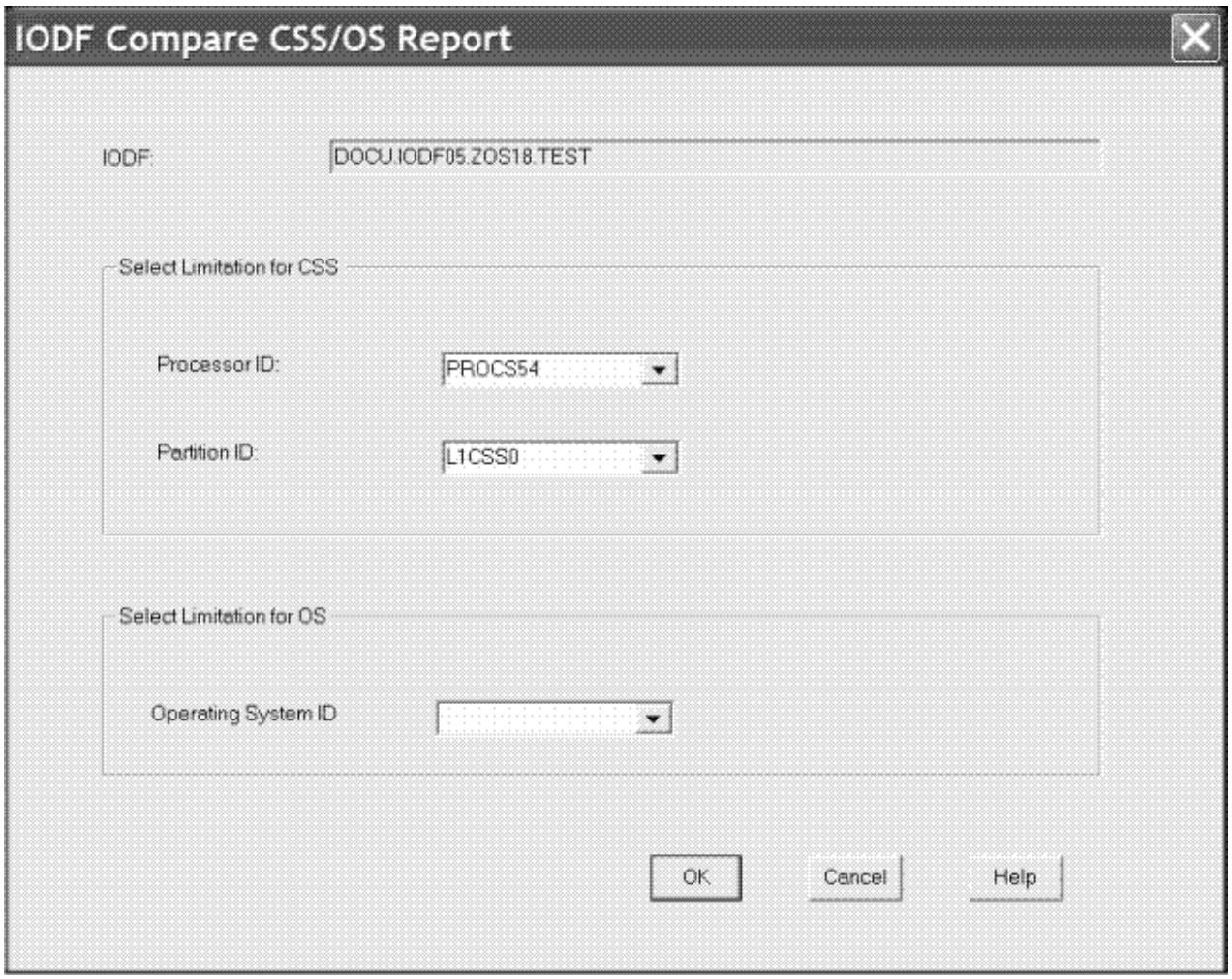


Figure 181. IODF Compare CSS/OS Reports

Specify your selections and press the **OK** button. HCM invokes another dialog that lets you specify whether you want to print all data or different data only.

Comparing HCM configurations

You can compare HCM configurations with respect to physical information. You can generate a configuration compare report using menu item ***File → Compare HCM Files...***. No configuration must currently be loaded, otherwise this selection is not enabled. HCM opens the following dialog:

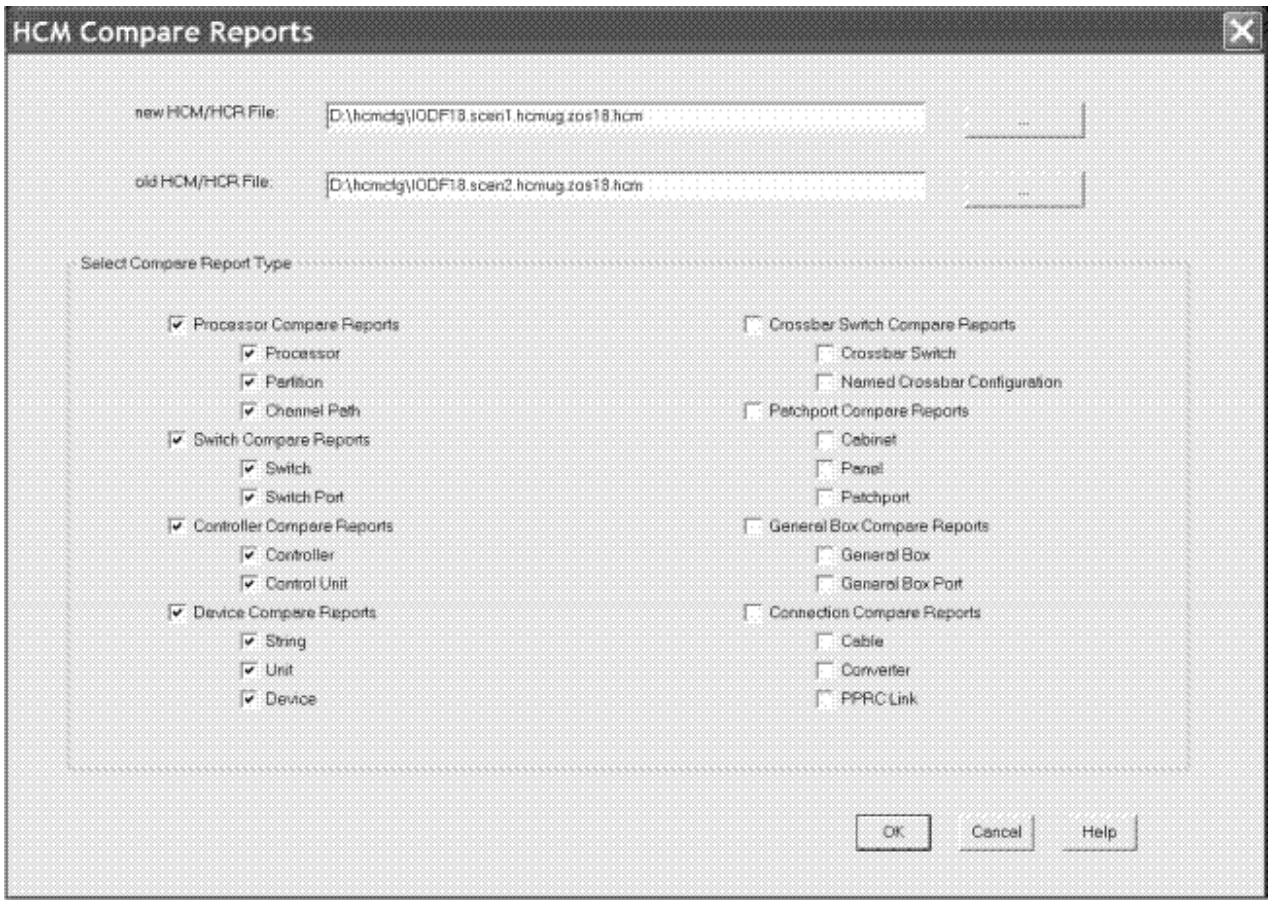


Figure 182. HCM Compare Reports

On this dialog, you can enter the configuration files and the types of reports that you want to be compared. After making your selections, press the **OK** button. HCM invokes another dialog that lets you specify options for printing the data resulting from report comparison (multiple selections are possible):

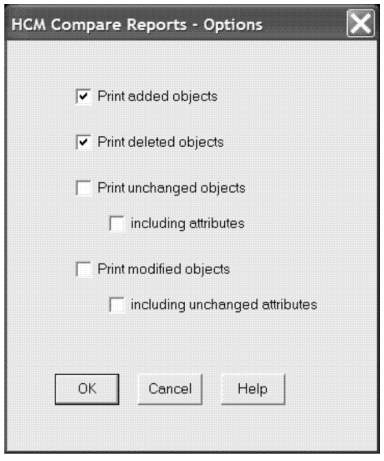


Figure 183. HCM Compare Reports - Options

Print added objects
to show objects in the new configuration that are not in the old configuration

Print deleted objects
to show objects in the old configuration that are not in the new configuration

Print unchanged objects
to show objects that are unchanged in both configurations (including or not including the attributes of these unchanged objects).

Print modified objects
to show objects that appear in both configurations with at least one modified attribute value. You can select to include the unchanged attributes values in the output, too, or to print the modified attribute values only.

The output of the compare report is displayed in a separate window and you can save it as plain text into a file. The layout of the HCM Compare Reports is the same as the layout of the HCD Compare Reports (see *z/OS HCD User's Guide*). The output of the compare report shows whether:

- an object exists in both configurations, or has been added or deleted
- object attributes are identical or modified in each configuration.

Note:

For comparison, both configurations are loaded in the same manner as during ***File → Open...*** processing. To avoid unexpected updates to either configuration during load, while you are connected to the host, it is recommended to run the comparison only when both configurations are in sync with their IODF. Alternatively, you can run the comparison in stand-alone mode.

Comparing HCM configurations is based on the contents of files created by the HCM **Export Data...** function (see Chapter 14, “Importing and exporting data,” on page 333). This means that each configuration is opened, exported to a temporary directory, and closed again. Then the exported files are compared. Therefore, messages EEQ0260E and EEQ0261E that may appear while opening/closing a configuration and exporting the involved data, can also appear while comparing HCM configurations. As a consequence, all data that cannot be exported into external files, cannot be compared. Just ignore the messages.

Chapter 12. 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:

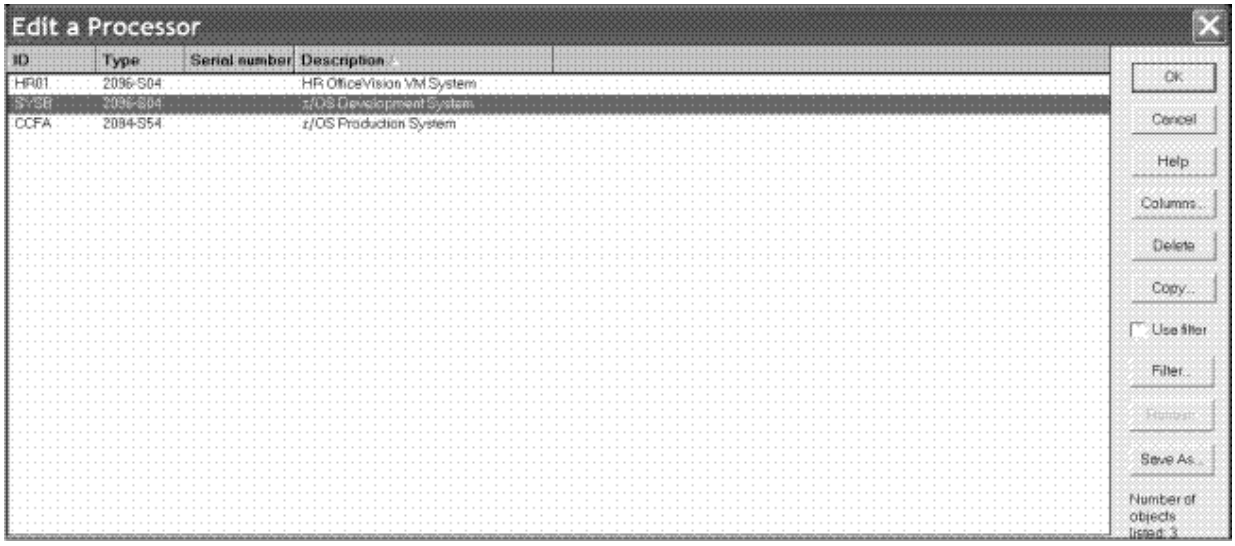
- "Scenario I – Adding and connecting new equipment" on page 285
- "Scenario II – Defining a CTC connection" on page 302
- "Scenario III – Defining a fiber cabling infrastructure" on page 311
- "Scenario IV – Configuration changes for installing a FICON bridge card in a 9032–5 ESCON director" on page 320
- "Scenario V – Establish a PPRC connection consisting of several PPRC links" on page 325

Scenario I – Adding and connecting new equipment

In this scenario we create new CHPIDs and strings for a processor with two partitions in channel subsystem 0 and connect them to each other through a new controller, a new FICON switch, and a patchport cabinet. An operation system configuration already exists.

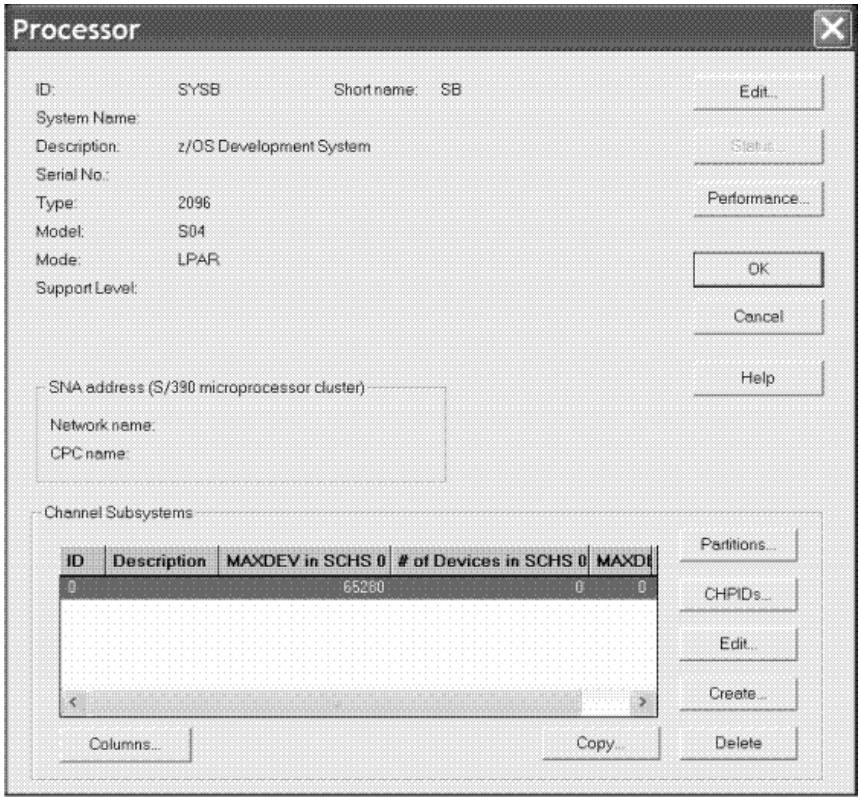
Add four new channels to an existing processor

1. Edit the processor by selecting **Edit --> Processor**. This opens the **Edit a Processor** dialog.

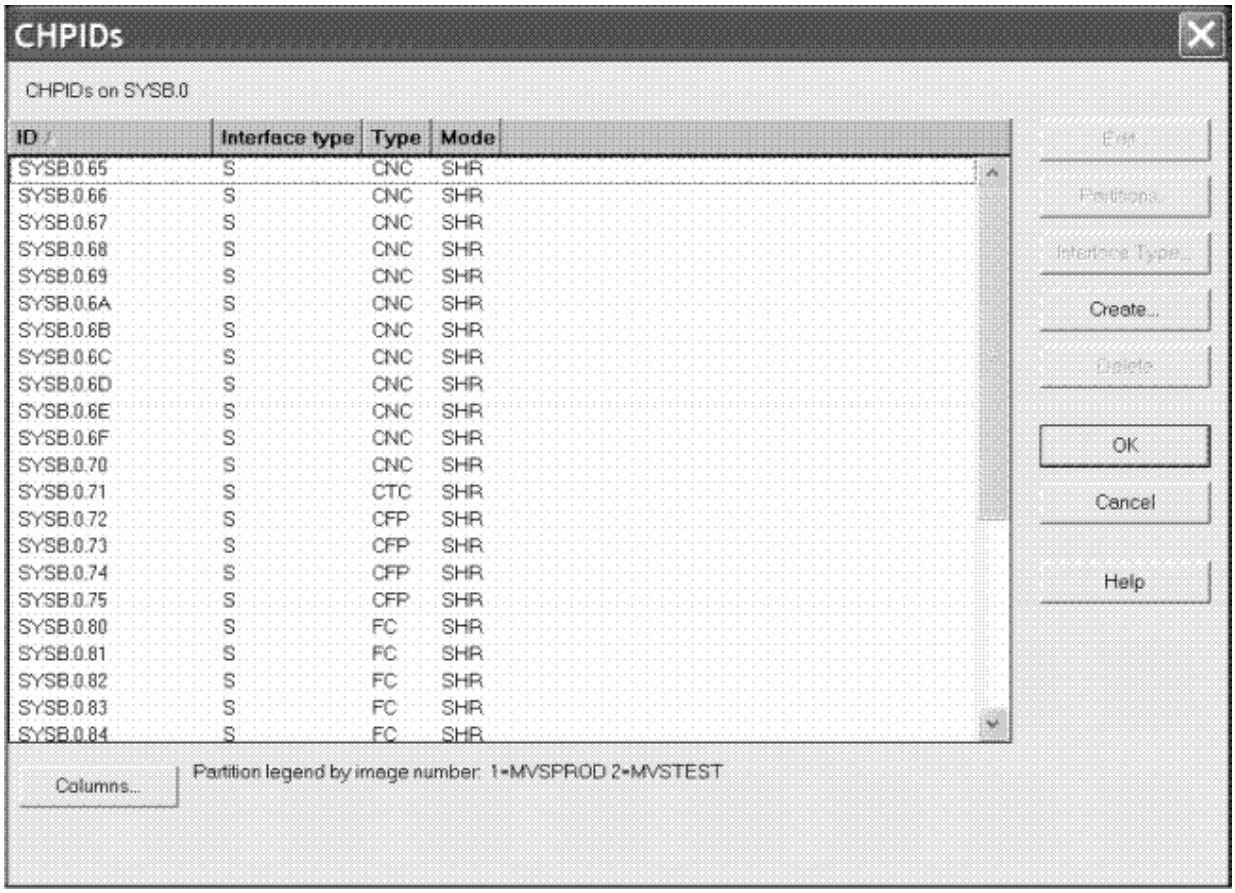


2. Choose the processor from the list (in our example, *SYSB*) and press **OK** or double-click on the processor in the list. This opens the **Processor** dialog.

© Copyright IBM Corp. 1995, 2021

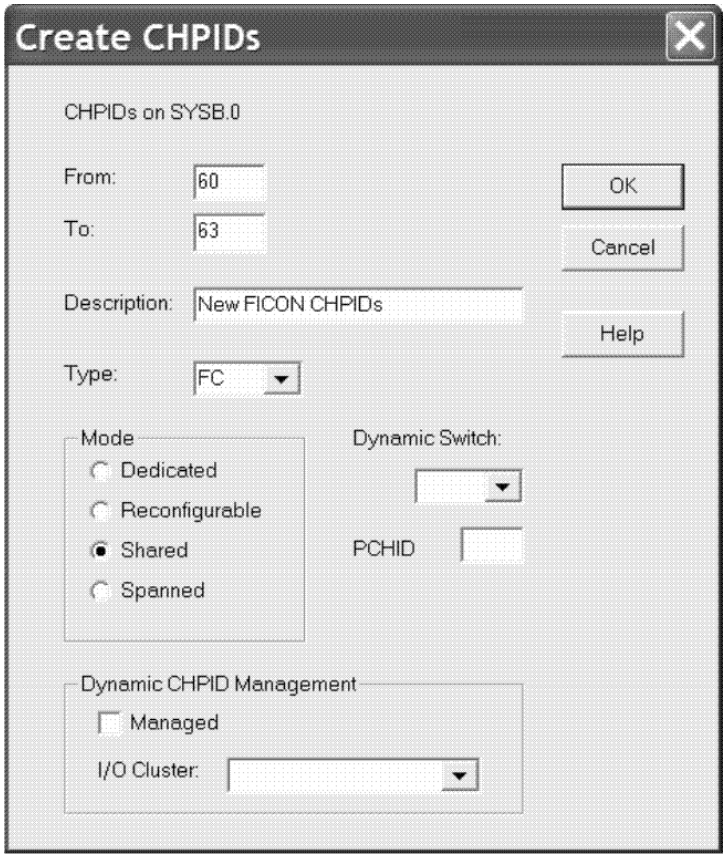


3. Click **CHPIDs...**. This opens the **CHPIDs** dialog.

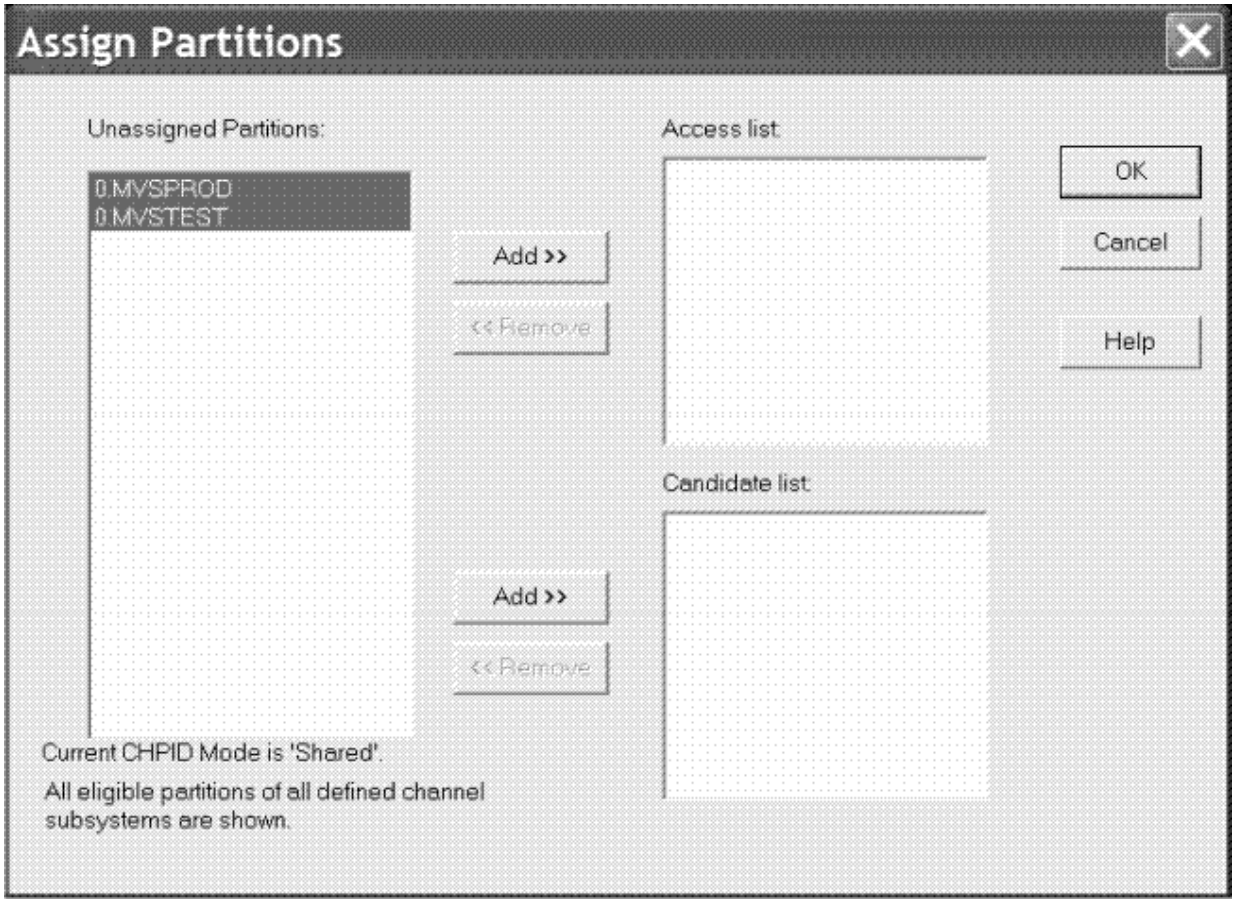


4. Click **Create...**. This opens the **Create CHPIDs** dialog.

5. Fill in the fields as shown, specifying appropriate CHPID numbers, to create four new shared (SHR) FC CHPIDs that can connect to a FICON switch:



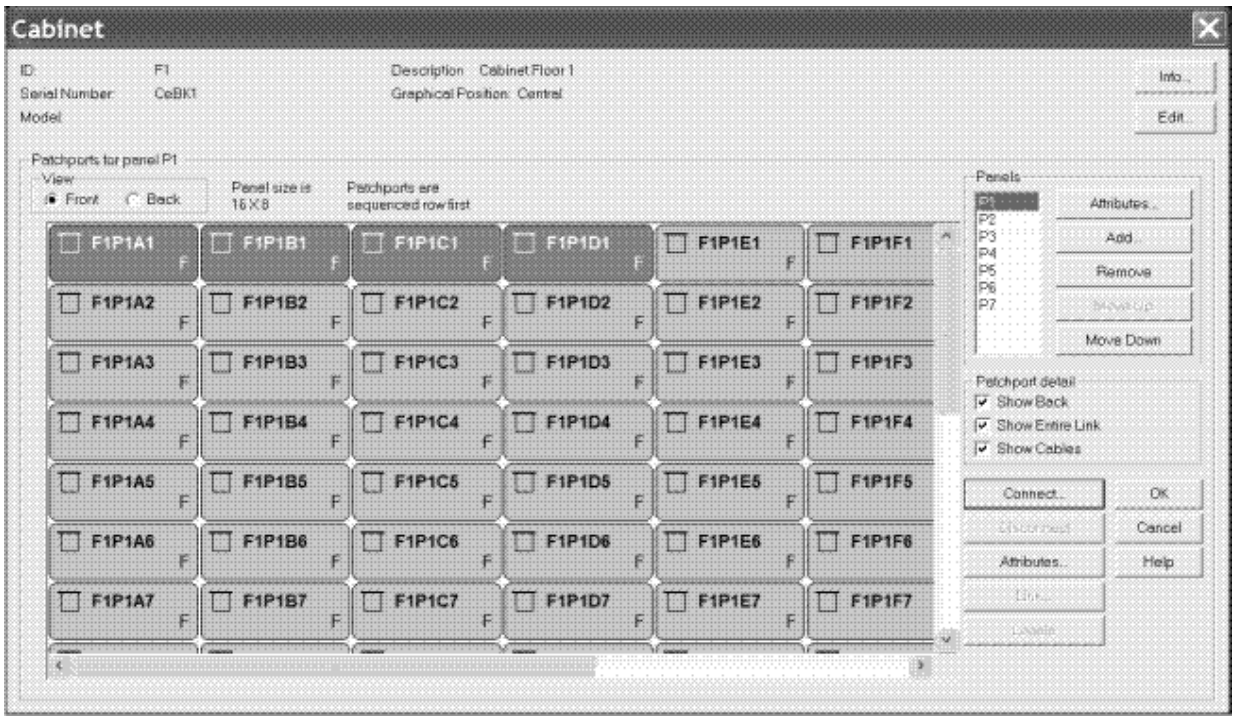
6. Click **OK**. This opens the **Assign Partitions** dialog to choose which partitions to assign to the access and candidate lists of the new CHPIDs.



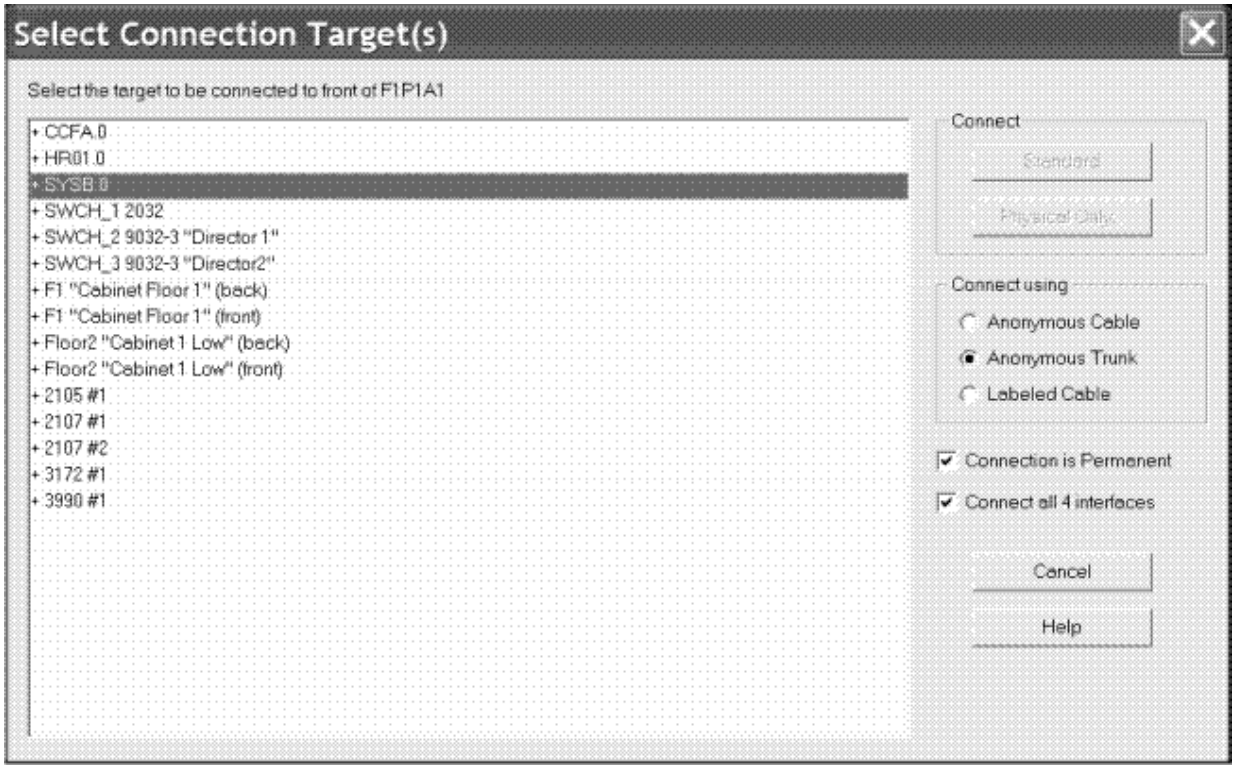
7. Select all of the partitions and click **Add>>** beside the access list. This adds the selected partitions to the access list of the four CHPIDs.
8. Click **OK** on all open dialogs to close them. The new CHPIDs have now been added.
9. Select **View --> Filter Diagram...** and select the **Show All 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*).



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**. HCM opens the **Select Connection Target(s)** dialog. In the listbox, double-click on the processor with the new CHPIDs (*SYSB.0*). This expands the list, showing all CHPIDs in the processor.

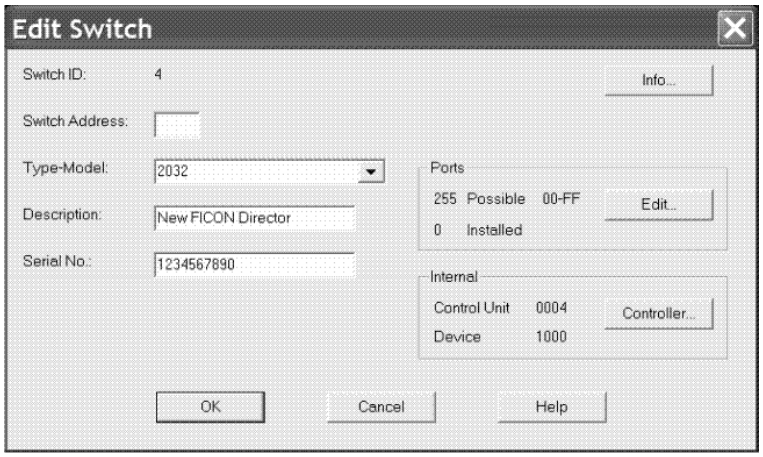


5. Select a CHPID and the ***Connect all 4 interfaces*** checkbox. Choose the type of connection you want to use and whether the connection should be permanent.
6. Press the **Physical Only...** button. The patchports are connected to four consecutive CHPIDs, starting with the one you have just selected.
7. Close all open dialogs.

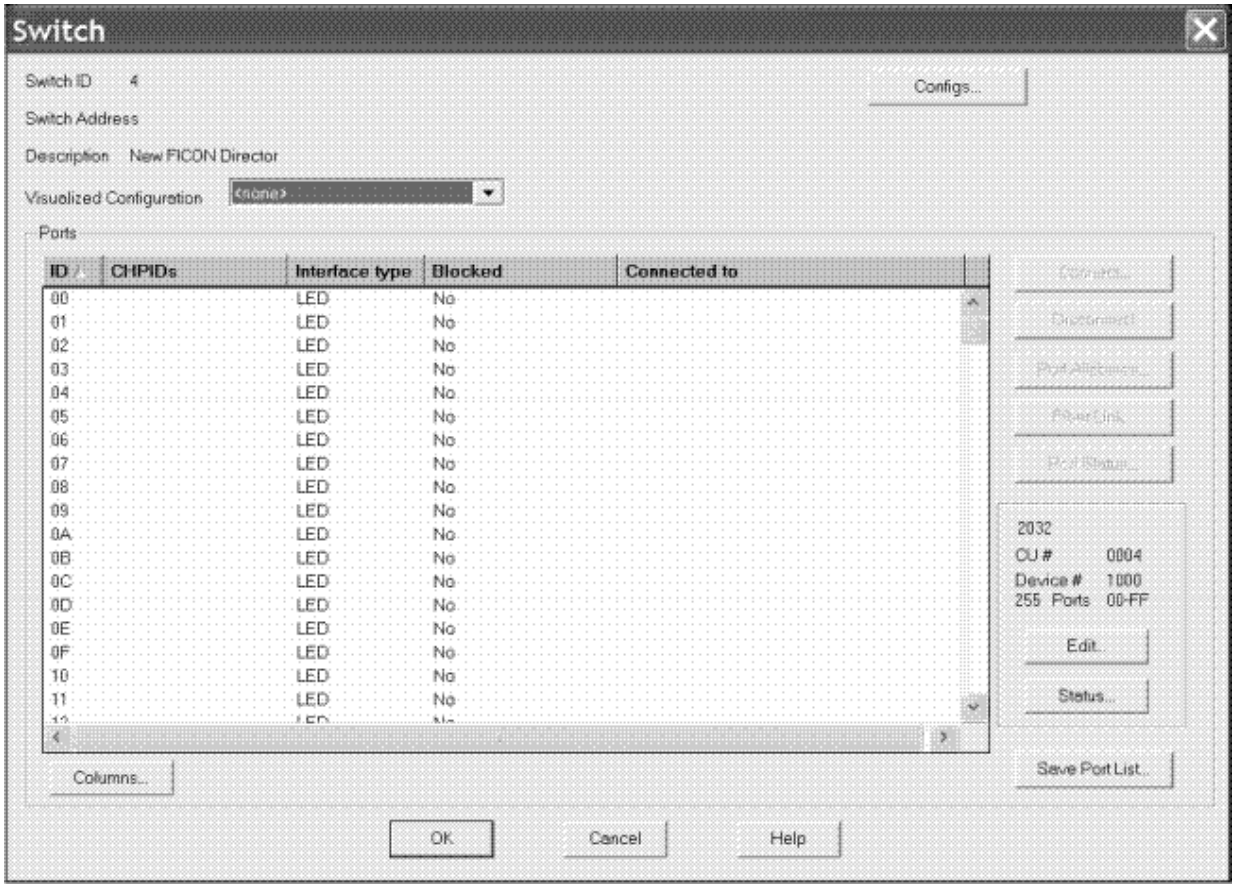
8. Select the **View --> Filter Diagram** menu item and select the **Objects Connected to Processors** and the **Show Patchports, General Box Ports, and Cables** checkbox. The patchports connected to the new CHPIDs appear in the diagram.

Add a new FICON 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 appears 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.



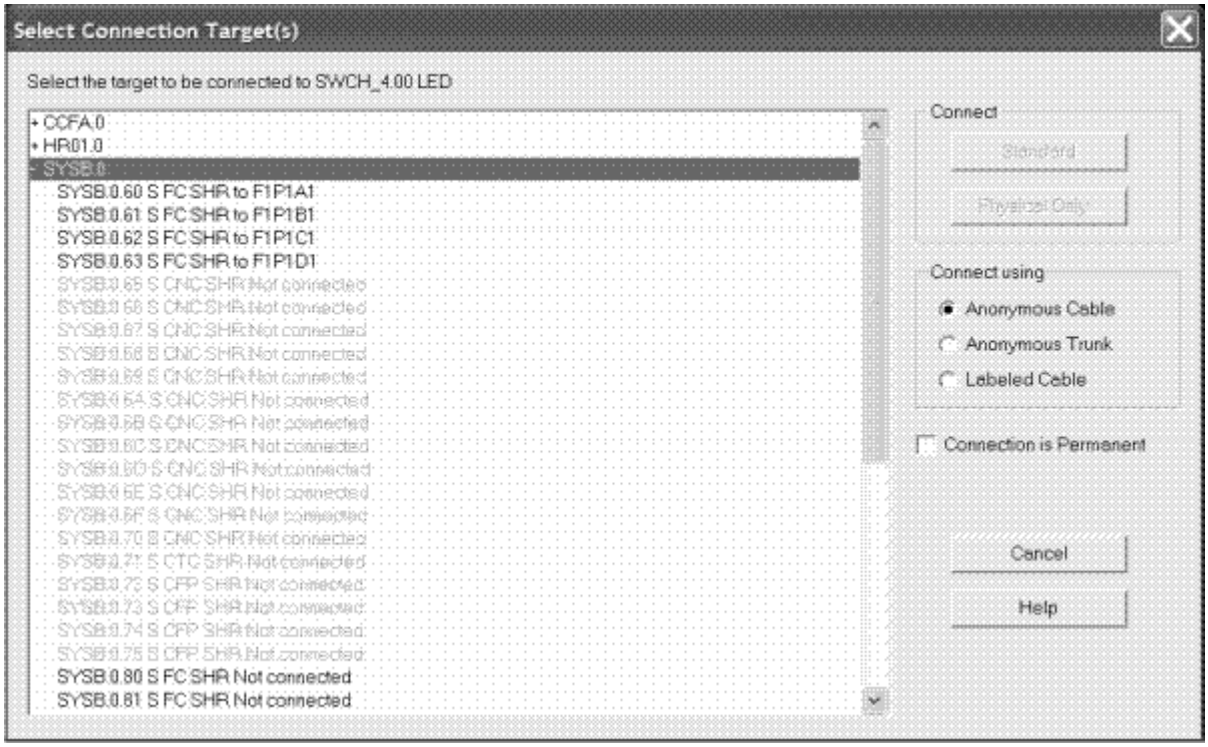
5. Click **Edit** in the **Ports** box. This opens the **Switch Ports** dialog.
6. Select the ports you want to install and click **Install**. The selected ports are now installed.
7. Click on the **Close** button, then **OK** on the **Edit Switch** dialog to return to the **Switch** dialog.



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 is 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 **Select Connection Target(s)** dialog.
2. In the list box, double-click on the processor (in our example SYSB.0) 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

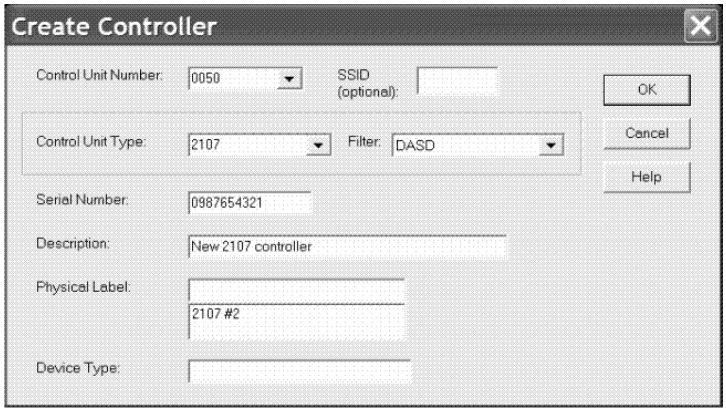
Follow the procedure above to connect another switch to the second new CHPID. You may have to create the second switch following the previous procedure. If there are no free installed switch ports you may have to mark some ports installed as described earlier.

Add a new controller

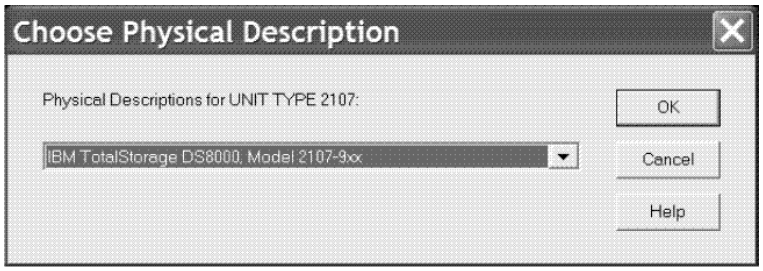
Note: Since HCM for z/OS 1.8, you could also use the ***Create I/O Subsystem*** wizard (see "Creating/Copying/Editing an I/O subsystem" on page 234) to achieve this task and the task described in "Add some strings to the new controller" on page 298 in contiguous steps and then establish the connections to CHPIDs and switches and define the cables for these connections as described in the rest of this scenario.

We are going to add a 2107 DASD subsystem.

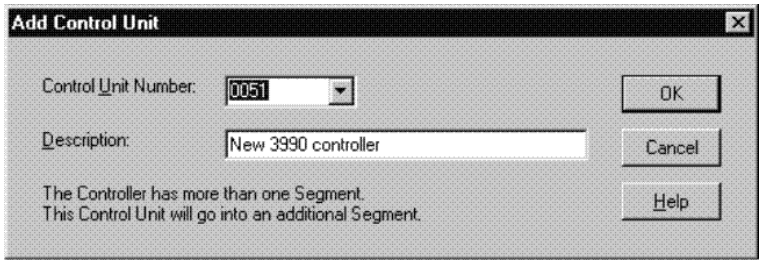
1. Select **Create --> Controller**. This opens the **Create Controller** dialog. Fill in the values as shown, specifying an unused control unit number, and click **OK**.



2. The **Choose Physical Description** dialog may open. 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. If you have defined your own PDF that allows two segments for your controller, 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.

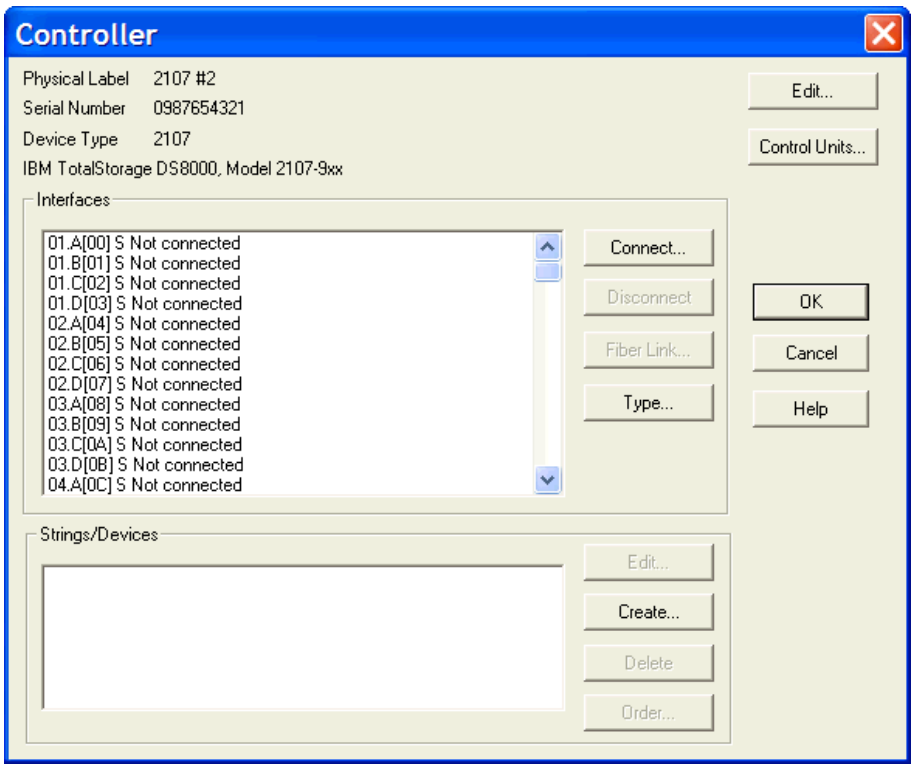
Connect the controller to CHPIDs

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”.

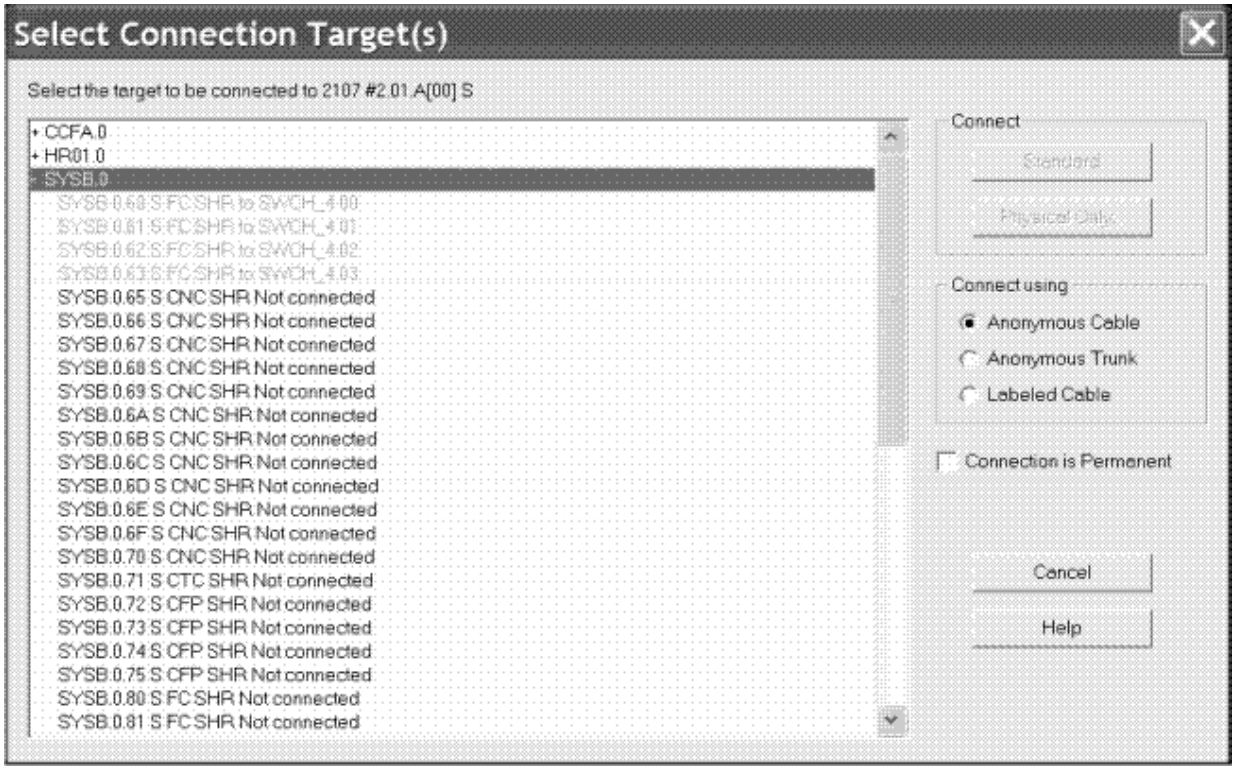
The advantage of connecting top-down is that you are given immediate feedback about available unit addresses when you connect strings to controllers. The advantage of connecting bottom-up is that when you connect a controller with strings to a CHPID, only the unit address ranges that are used by the connected devices are defined for the controller, instead of the default unit address range of 00 through 64.

We are connecting top-down, starting with connecting one channel interface on each segment of the new controller to a CHPID.

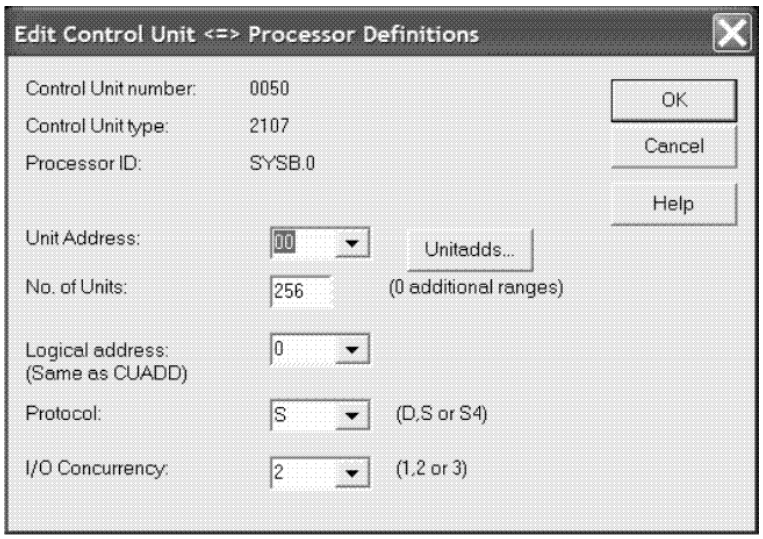
1. Edit the new controller by either selecting **Edit --> Controller** and choosing the controller from the list or by double- clicking on the controller in the diagram. This opens the **Controller** dialog.



2. Select a free channel interface on the first segment (for example, 01.A) and click **Connect**. This opens the **Select Connection Target(s)** dialog.



3. Choose the desired processor (in our example SYSB.0) from the list and double-click it to expand it. This lists all of the CHPIDs in the processor.
4. Select a SHR FC 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 following dialog opens:



In this dialog, you can edit the attributes for the logical definitions between the selected control unit and one or more processors. In our scenario, you can leave the values unchanged and press the **OK** button.

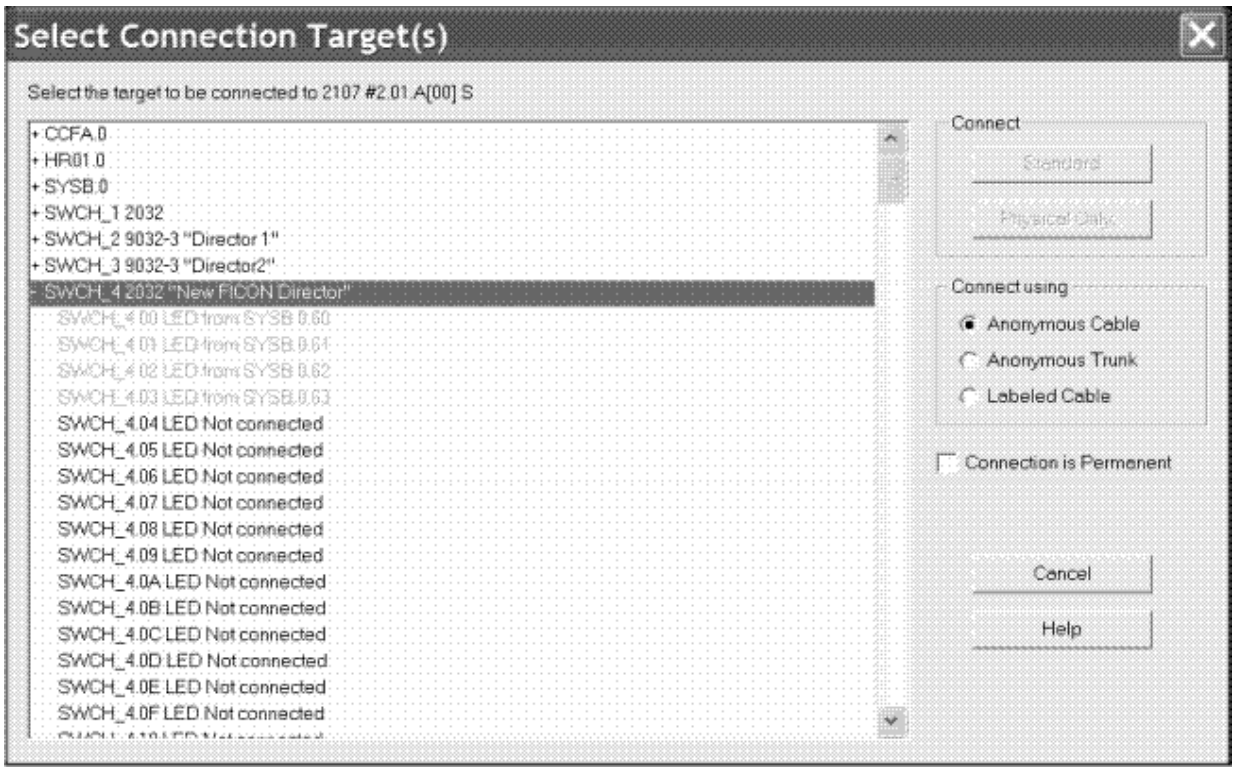
The connection is now made including all definitions between the control unit and the CHPID and its processor. Had there been devices connected to the control unit, definitions would also have been made between the devices and the partitions and the processor.

5. You are now back in the **Controller** dialog. If you have defined your own PDF that allows two segments for your controller, select a free channel interface on the second segment and connect it to another SHR FC CHPID following the previous procedure.
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.

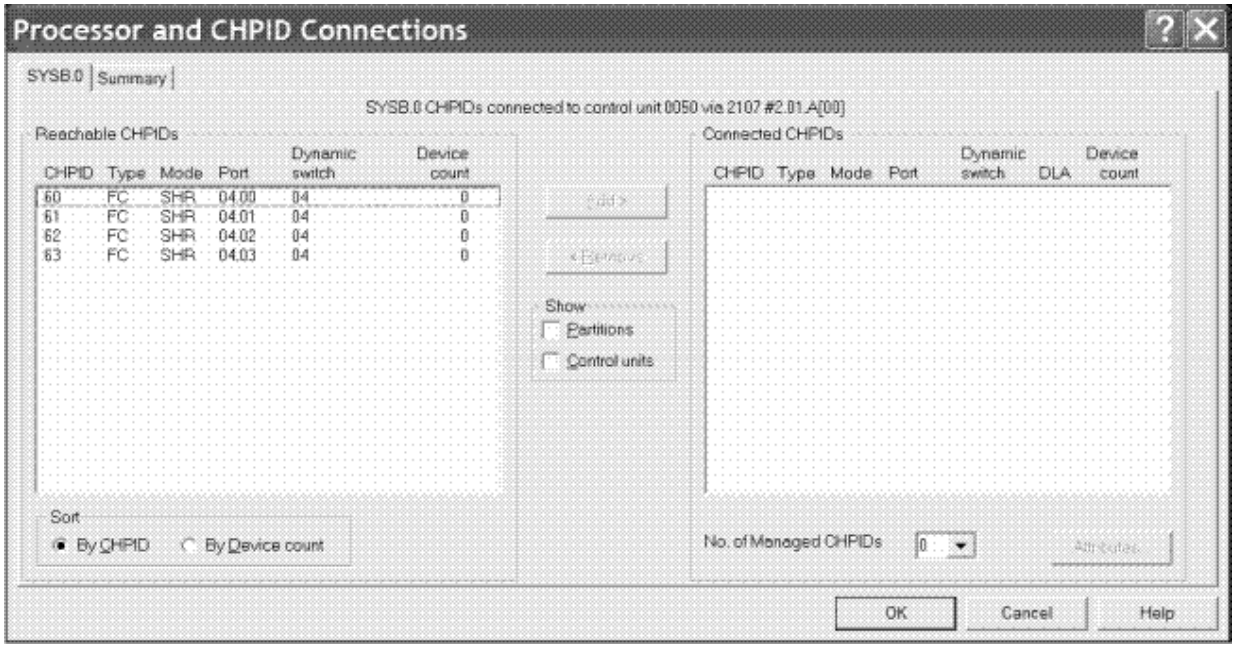
Connect the controller to the FICON switches

We connect a second channel interface of the new controller through a switch and on to the new CHPIDs.

1. Open the **Controller** dialog for the new controller as before.
2. Select a free channel interface 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.



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.

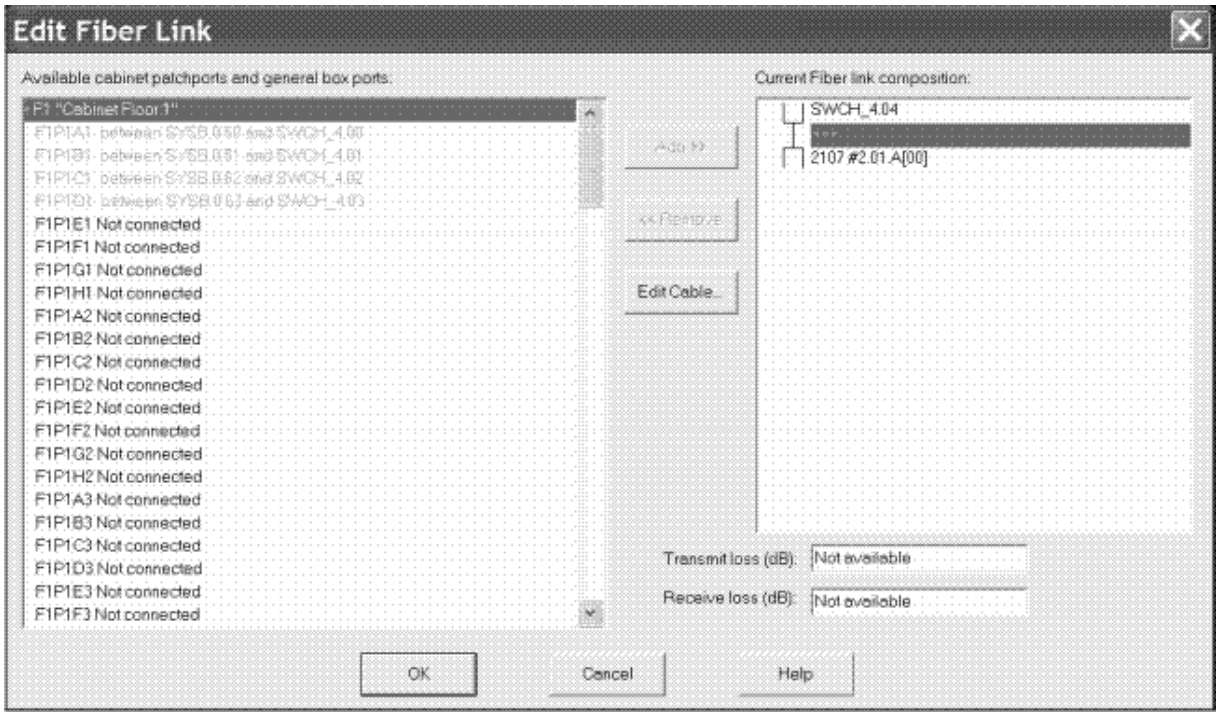
7. You are now in the **Controller** dialog. Click **OK** on all open dialogs to close them. The connections between the controller and the switches appear on the diagram. When the controller is selected, the full connection is highlighted from the controller through the switches and the patchports, to the CHPIDs, and finally to the partitions.

Define the fiber links for the connections

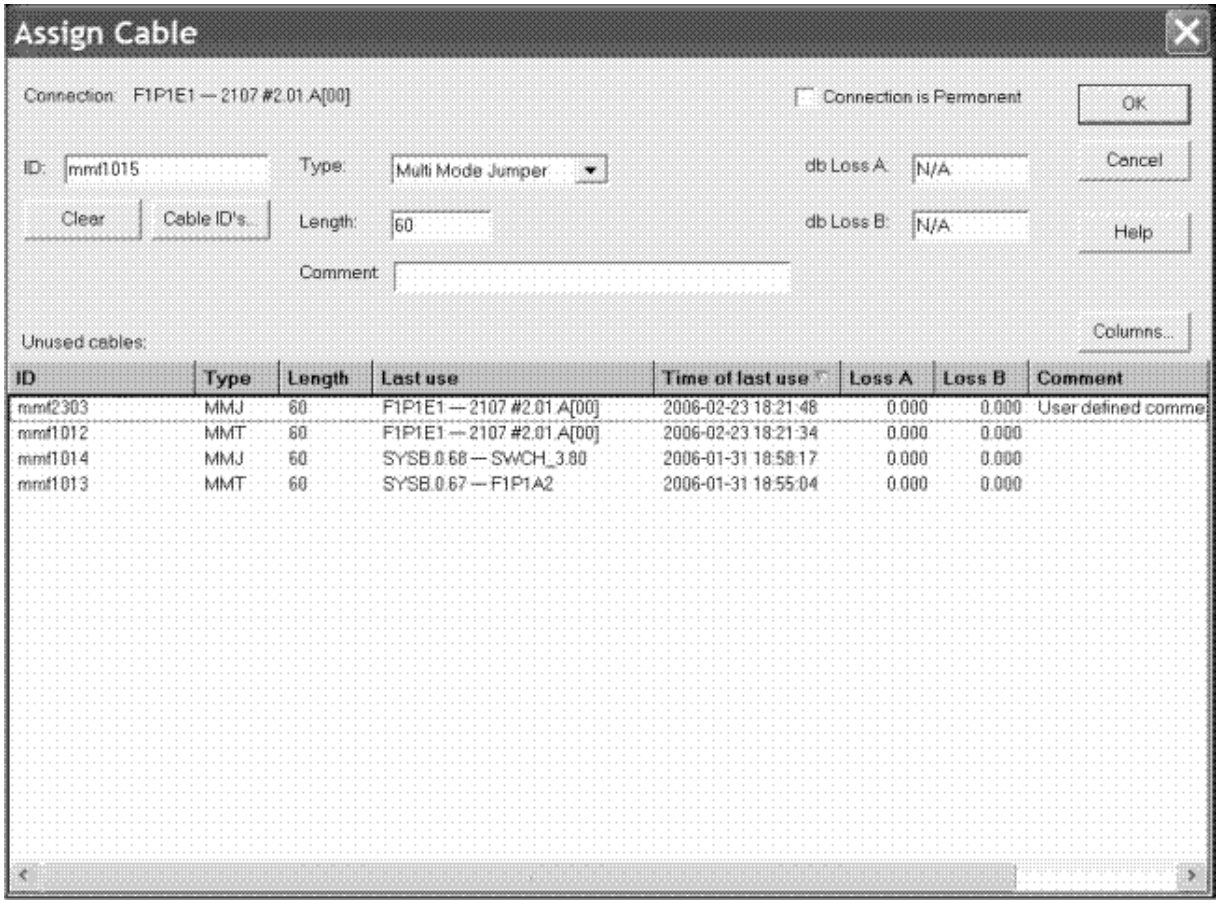
When a connection is made between two pieces of equipment no cable is initially assigned to the connection. You can assign cables, complete with labels, types, and lengths, to the connections in your configuration. This information can then appear on the diagram and can be printed in reports. If the connection is between two fiber transceivers, you can add patchports to the fiber link.

We route one of the connections we just made between the new controller and a switch, passing through a patchport cabinet, and label the involved cables.

1. Edit the new controller.
2. In the **Controller** dialog, select (do not double-click) the channel interface that is connected a switch. 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.



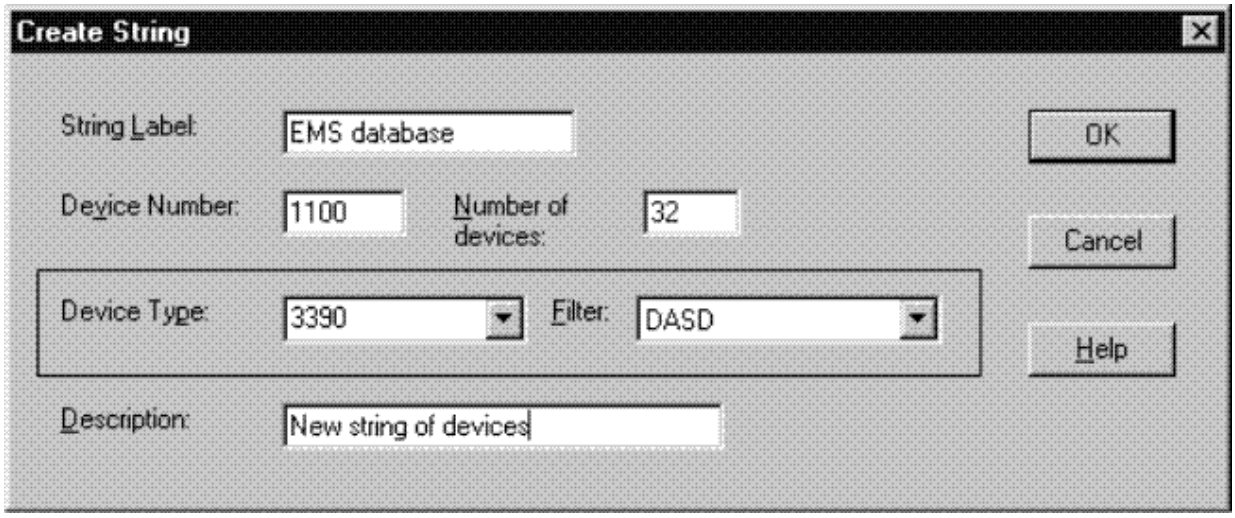
6. Select a free patchport and click **Add>>**.
7. The **Add Patchport (General Box Port) to Fiber Link** dialog appears. You can decide whether you want to connect the cables to the front or to the back side of the patchport. This inserts the patchport into the fiber link such that there are now two unlabeled cables on either side of it. Press **OK**.
8. Now we label one of the cables. Select the "- - -" between the controller and the patchport and click **Edit Cable**. This opens the **Assign Cable** dialog.



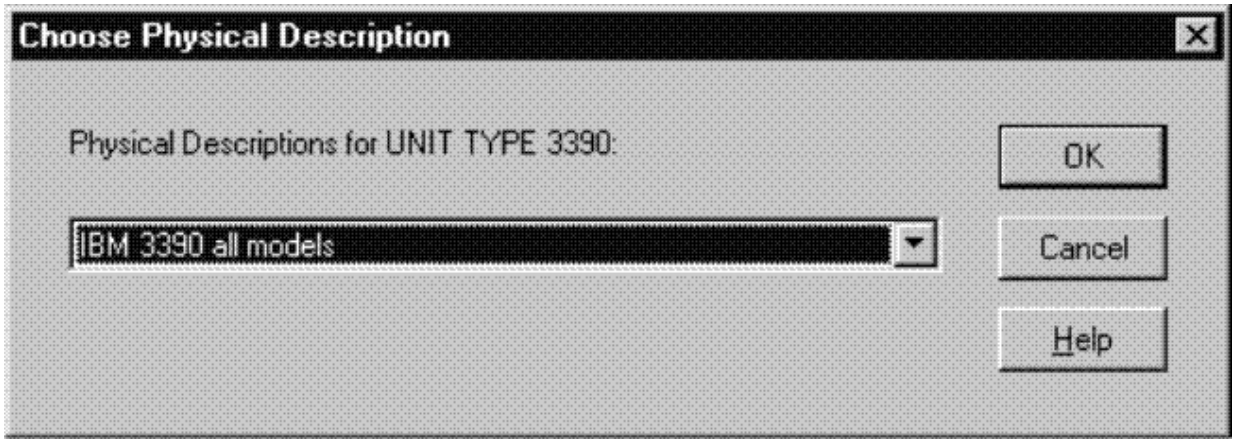
9. Fill in the cable's ID, type, length, and optionally the Comment field, or select an existing cable from the unused cables list. Then, click **OK**. The cable's new label now appears in the **Edit Fiber Link** dialog.
10. Label the other cable in the same way.
11. Click **OK** to return to the **Controller** 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 now appear on the diagram.

Add some strings to the new controller

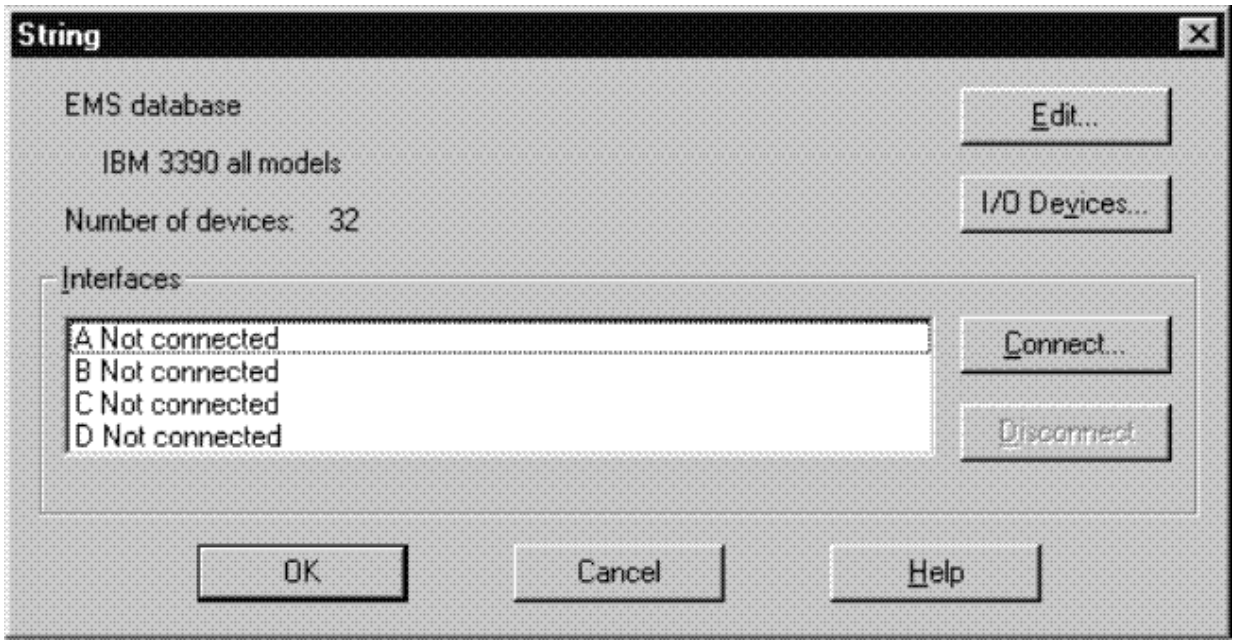
1. Select **Create --> String**. This opens the **Create String** dialog.



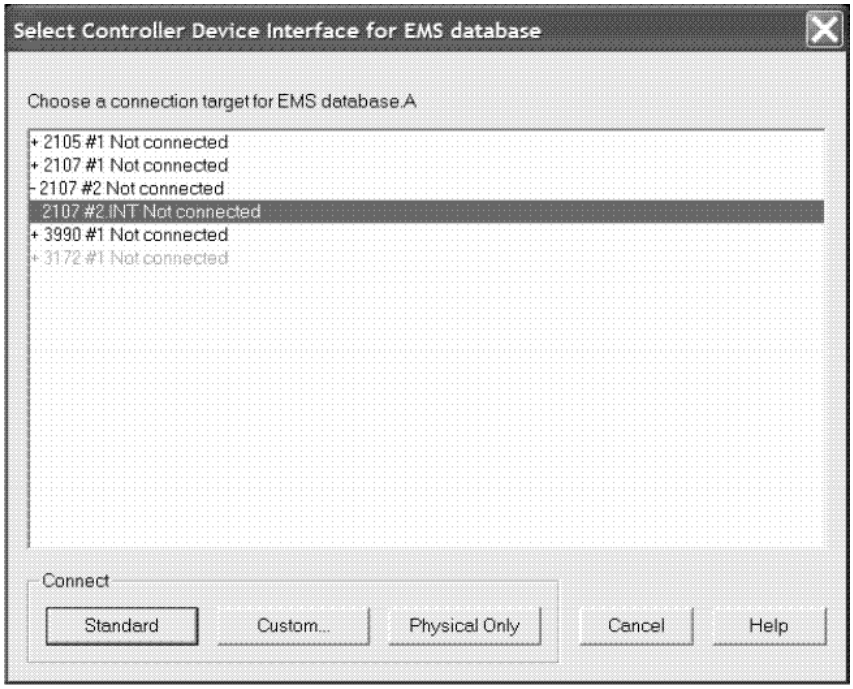
2. Fill in the **Create String** dialog as shown, specifying an appropriate device number.
3. Since there is more than one physical description for this unit type, the **Choose Physical Description** dialog is opened. Choose the appropriate physical description and click **OK**.



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.)



5. Select the first string interface (A) and click **Connect**. This opens the **Select Controller Device Interface** dialog.



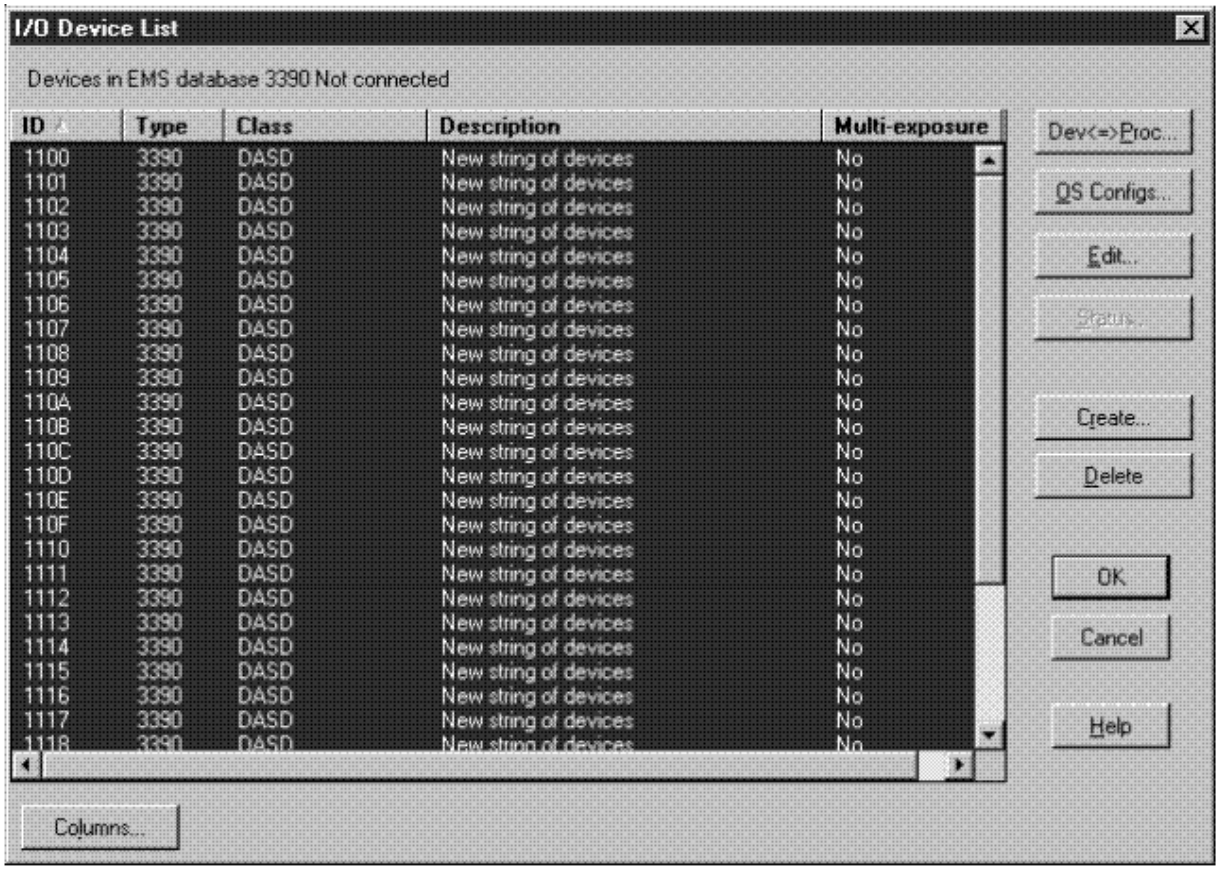
6. Double-click on the new controller (2107 #2) and choose the storage processor (a so-called “device interface” in HCM) to connect to and double-click on it.

   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 opens 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.

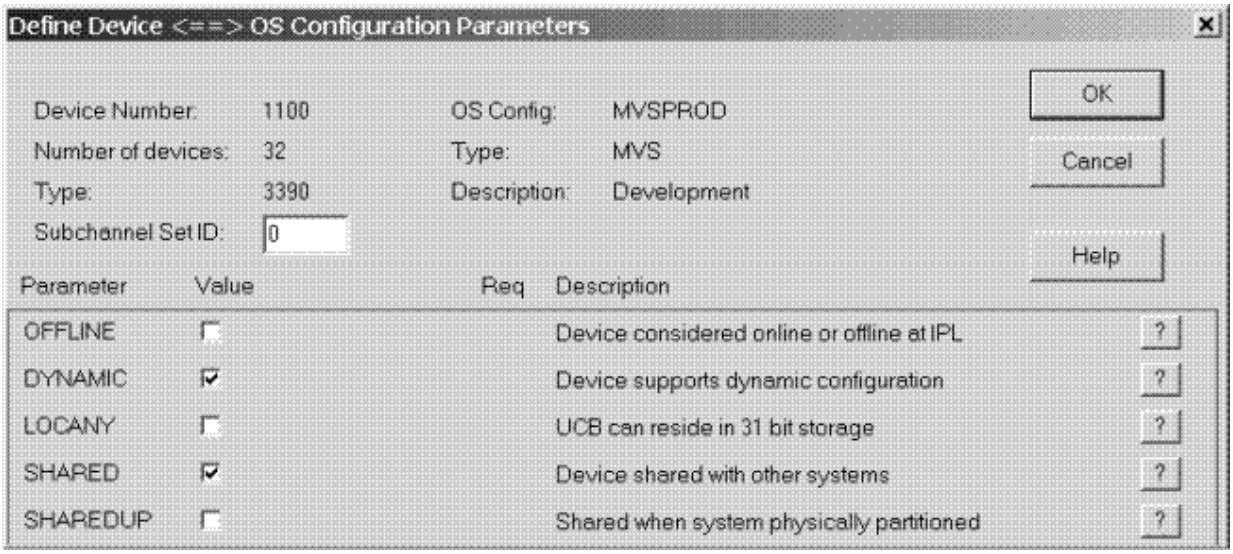
Define the devices to the operating system

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:

- ***SMP Processor PROC1*** (type 2064 with partitions LPAR11 and LPAR12) with CHPID:
  - 20 CTC SHR (LPAR11 in the access list)
- ***SMP Processor PROC2*** (type 2064 with partitions LPAR21 and LPAR22) with CHPIDs:
  - 20 CTC SHR (both partitions in the access list)
  - 21 CNC SHR (both partitions in the access list)
- ***XMP processor PROC3*** (with LCSS 0 and partition LPAR31) with CHPID:
  - 21 CNC SHR (LPAR31 in the access list)

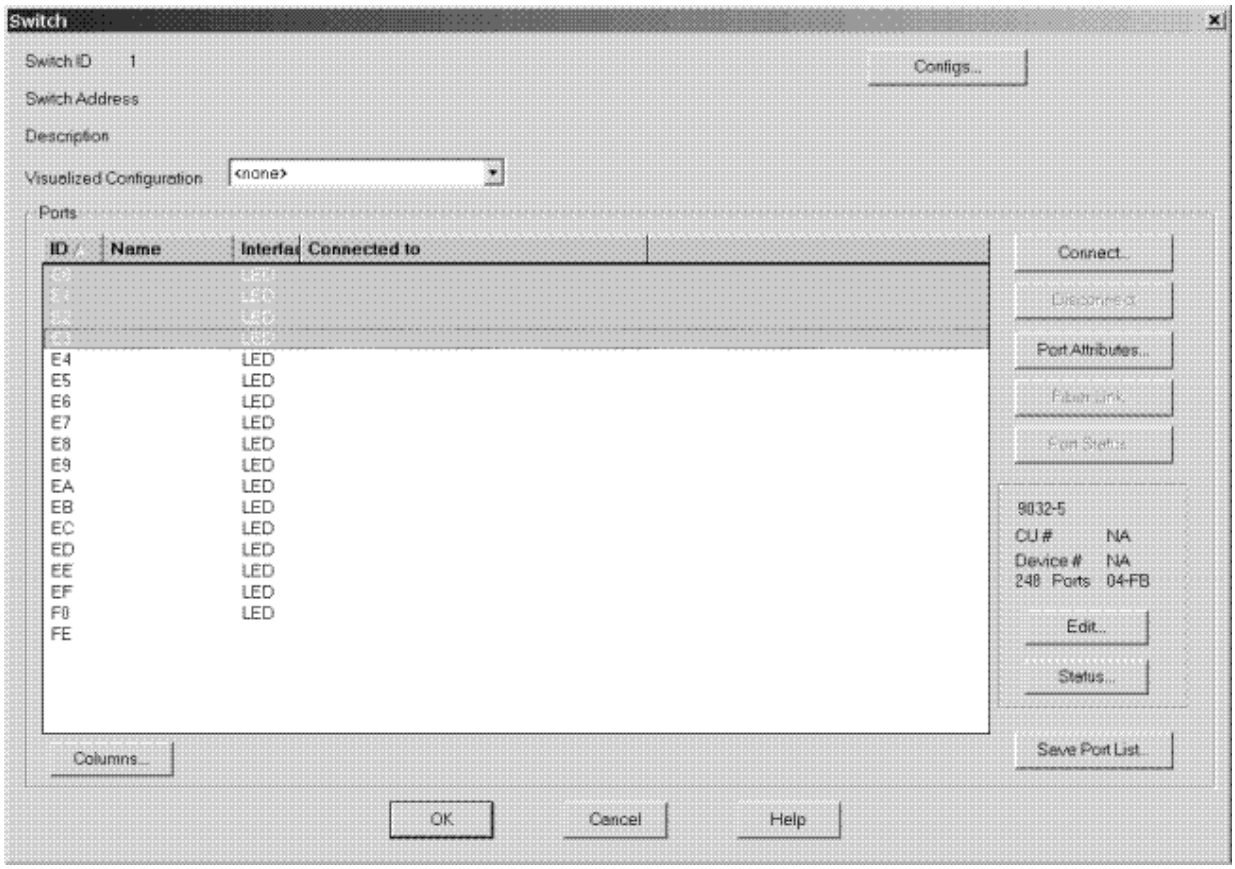
We also assume that an operating system exists (OS1 of type MVS) and that an ESCON Director 01 (9032-5) exists with ports E0-E3 free.

We create four CTC connections:

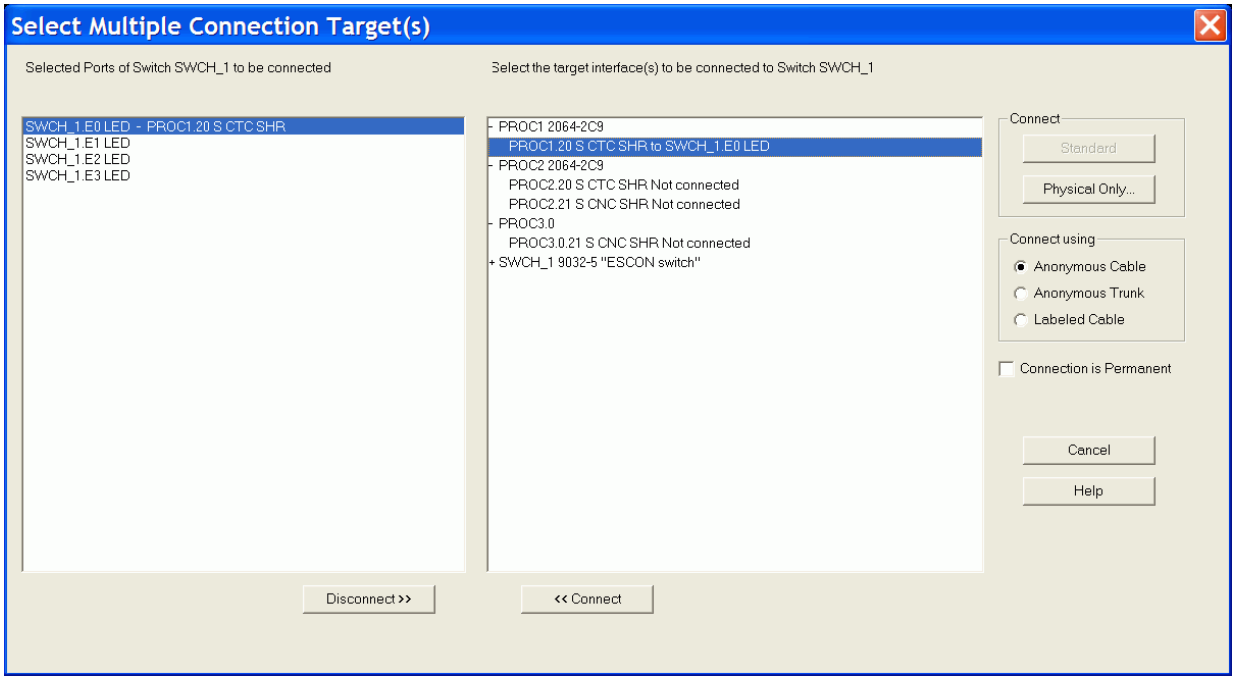
PROC1.20 CTC to PROC2.21 CNC
PROC2.20 CTC to PROC2.21 CNC
PROC1.20 CTC to PROC3.0.21 CNC
PROC2.20 CTC to PROC3.0.21 CNC

Connect a switch to the CHPIDs

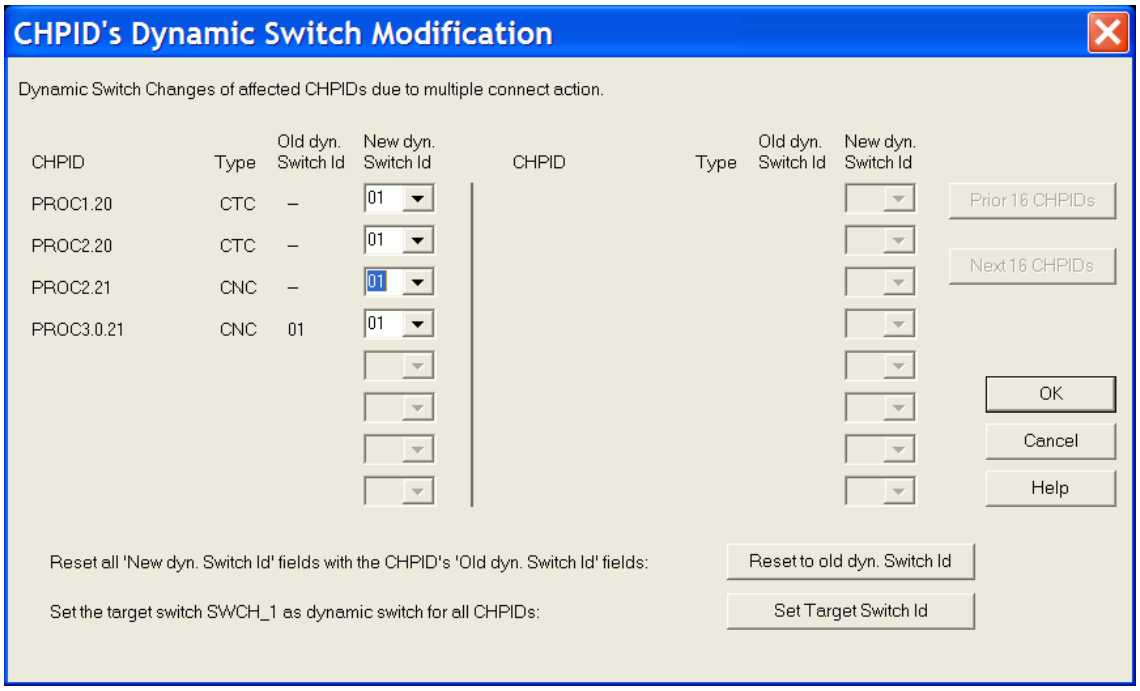
1. Edit the switch by either selecting **Switch** from the **Edit** menu and choosing the switch from the list or by double-clicking on the switch in the diagram. This opens the **Switch** dialog.



2. Select four free ports (E0 through E3) and click **Connect**. This opens the **Select Multiple Connection Target(s)** dialog, allowing you to select the connection target interfaces.



3. In the right list box of the **Select Multiple Connection Target(s)** dialog, double-click on the + signs beside processors PROC1, PROC2 and PROC3 to show the contained CHPIDs in these processors. Then from PROC1, select this one: CHPID PROC1.20 CTC SHR.
4. In the left list box, select port E0 and press the enabled ***<<Connect*** button at the bottom of the window. This establishes the requested connection between port and CHPID.
5. Repeat steps “3” on page 303 and “4” on page 303 to connect CHPIDs PROC2.20, PROC2.21 and PROC3.0.21 to the switch ports E1 – E3. Then press button **Physical Only ...**.
6. In the **CHPID's Dynamic Switch Modification** dialog, enter 01 into the ***New dyn. Switch id*** fields for all CHPIDs and then press **OK**.



7. Close the **Switch** dialog to return to the diagram. Now select **Filter Diagram...** from the **View** menu. Enable the following check boxes:
   - **CTC Connections**

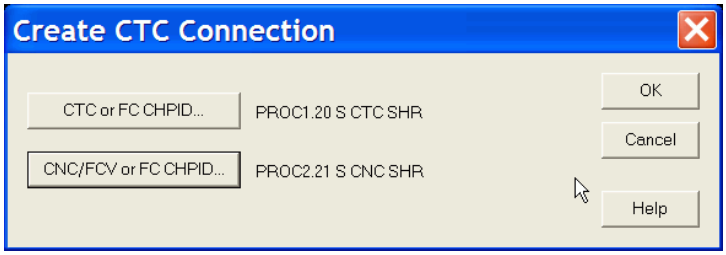
- **Objects with physical-only connections**
- **Show Processors**
- **Show All CHPIDs**

The new connections appear in the diagram.

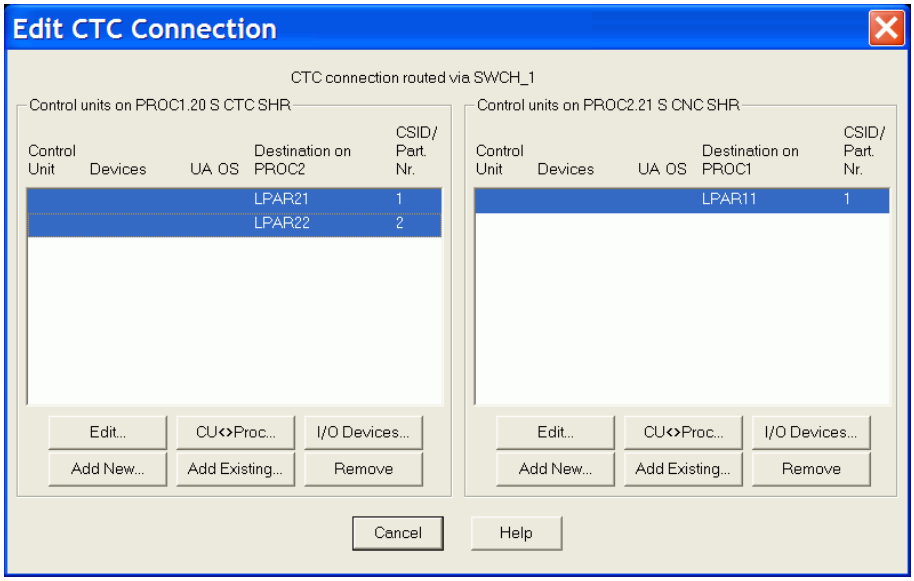
Create the first CTC connection

HCM considers a CTC connection as a single entity comprising selected control units and I/O devices connected to the CHPIDs.

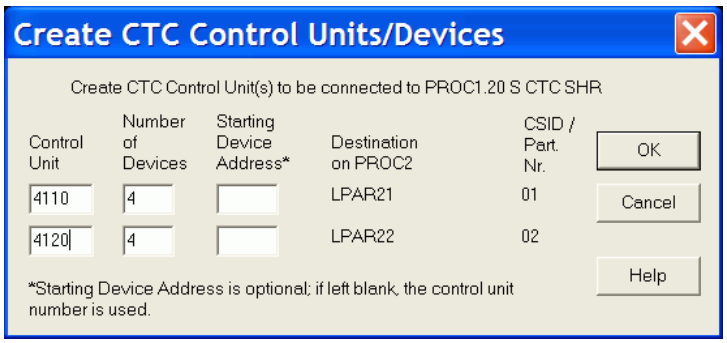
1. Select **CTC connection...** from the **Create** menu. This opens the **Create CTC Connection** dialog.
2. Click the **CTC or FC CHPID** button, and from the emerging **Select CHPID for CTC connection** dialog, double-click on the + sign besides PROC1, then select PROC1.20 S CTC SHR and finally press **OK**. Even though the CHPIDs are connected to switch ports, we select the CHPIDs that are to be connected. Click the second **CNC/FCV or FC CHPID** button and select PROC2.21 CNC SHR.



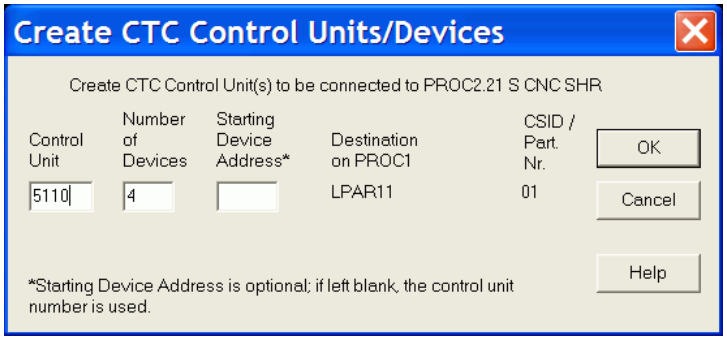
3. Click **OK** in the **Create CTC Connection** dialog. The **Edit CTC Connection** dialog appears to allow you to define the control units that are part of the connection.



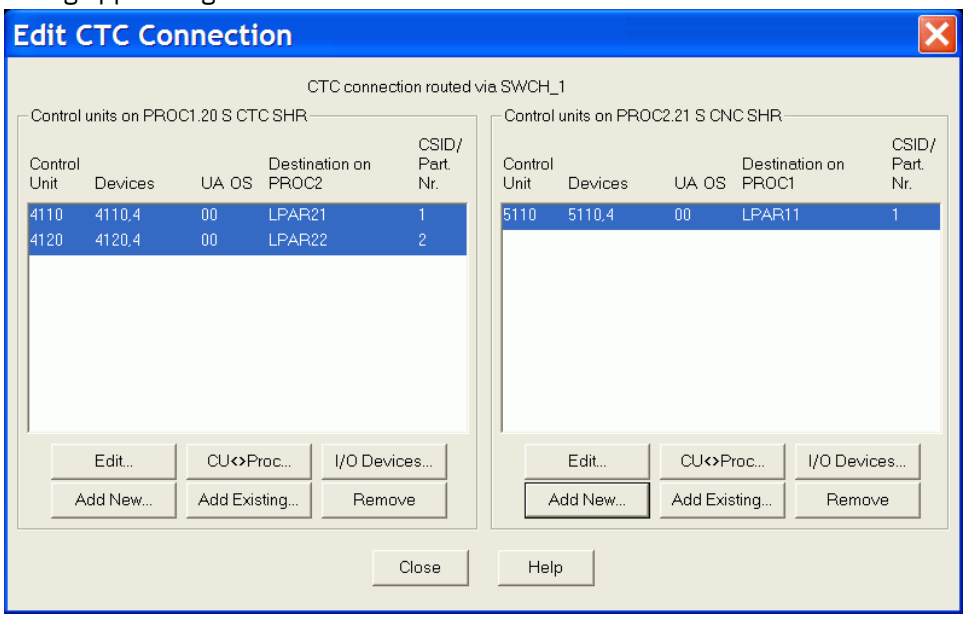
4. Click the **Add New...** button on the left-hand side to create the control units and devices that are connected to PROC1.20 CTC to represent the images available via PROC2.21 CNC.
5. Enter control unit number 4110 to represent LPAR11 and control unit number 4120 to represent LPAR12. Leave the other controls as they appear and click **OK**. This creates 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 to represent the two partitions.



6. Click the **Add New...** button on the right-hand side of the **Edit CTC Connection** dialog to create the control unit and devices that are connected to PROC2.21 CNC to represent the images available via PROC1.20 CTC.

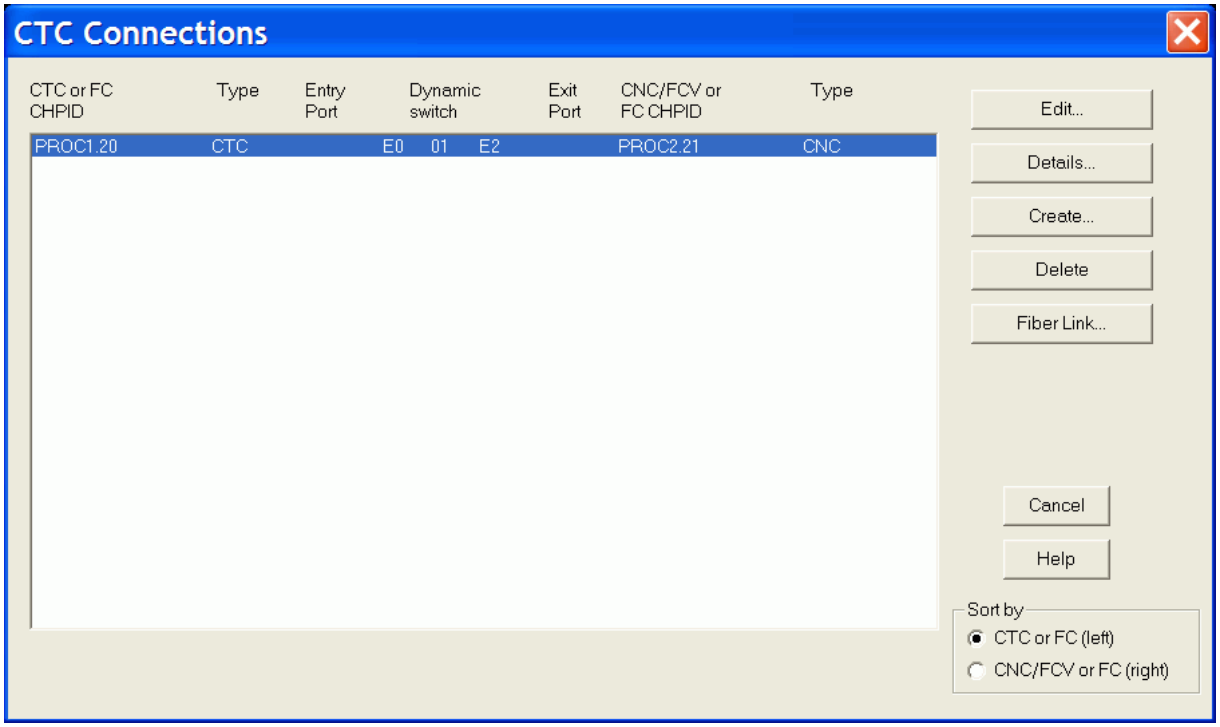


7. Enter control unit numbers 5110 to represent the image and click **OK**. The **Edit CTC Connection** dialog appears again as follows:

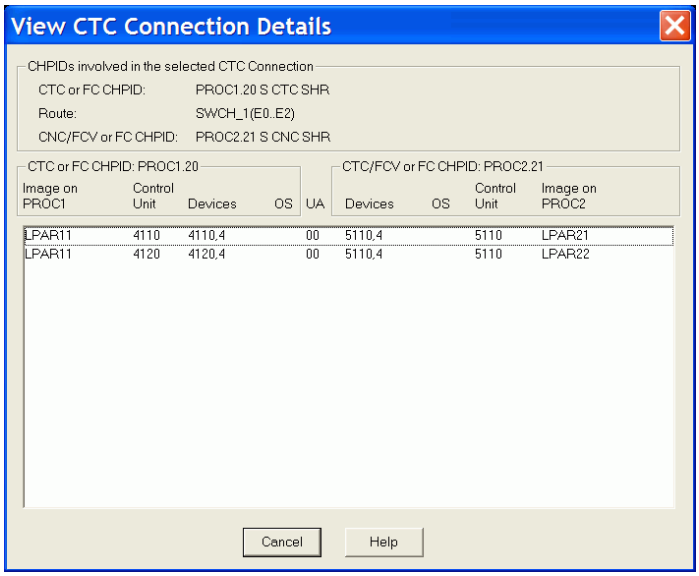


This dialog shows that the connection is fully specified with all images duly represented by control units.

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 PROC2.21 CNC appear in the dialog.



9. Select the entry in the list box and click **Details...** to open the **View CTC Connection Details** dialog.



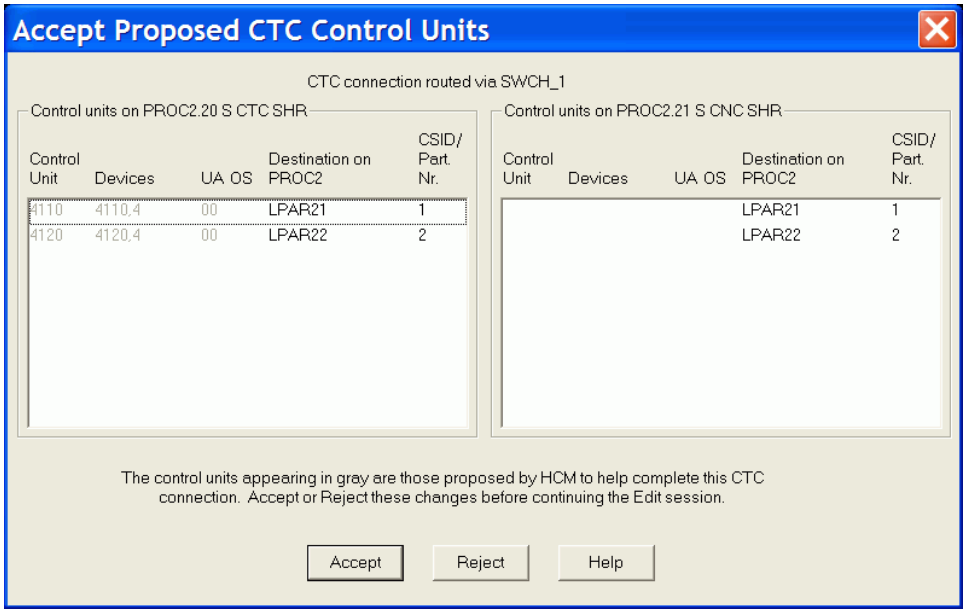
10. This gives another view of the connection. The partition image LPAR11 on PROC1 can communicate with partitions LPAR21 and LPAR22 on PROC2.
11. Click **Cancel** to return to the **CTC Connections** dialog.

Create the second CTC connection

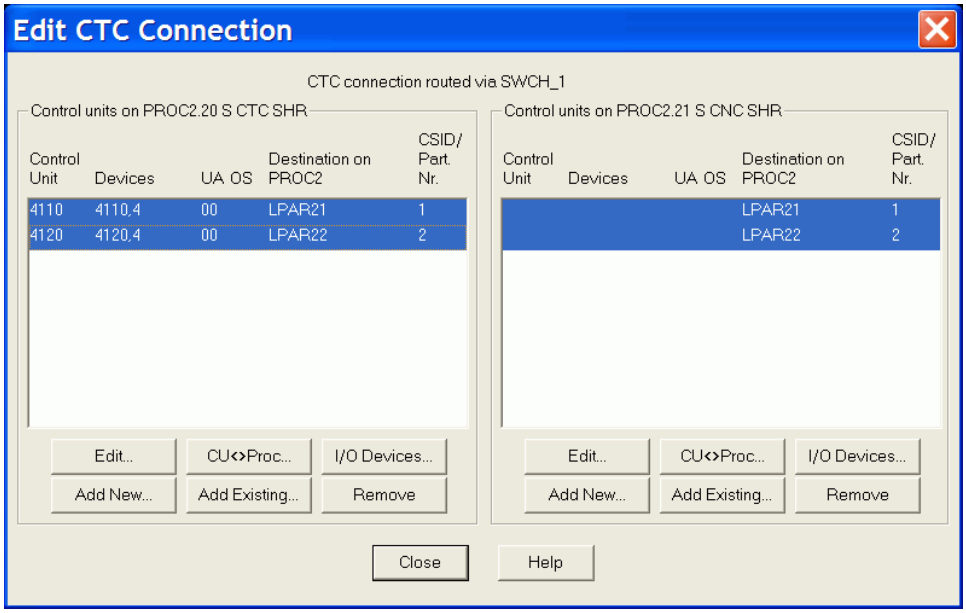
The second connection is between PROC2.20 CTC and PROC2.21 CNC. In this case, we have already created control units and devices that may be used to represent partitions LPAR21 and LPAR22 via PROC2.21 CNC; we need only connect them appropriately to PROC2.20 CTC . We will have to create a control unit and 4 devices to represent 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 SHR.
3. Click the second **CNC/FCV or FC CHPID** button and select PROC2.21 CNC SHR.

4. Click **OK**. This displays the **Accept Proposed CTC Control Units** dialog which proposes control units 4110 and 4120 to be connected to PROC2.20 CTC to represent LPAR21 and LPAR22, respectively.



5. Click **Accept** to connect the control units to PROC2.20. The **Edit CTC Connection** dialog appears, showing that the CTC side has been defined. Note that the CNC side of the dialog shows that there are two images to be represented. This is because PROC2.20 CTC has two partitions in its access list, LPAR21 and LPAR22. As yet, there are no control units to represent these partitions.



6. Click the **Add New...** button on the right hand side to create the control units and devices that are connected to PROC2.21 CNC to represent the partitions (LPAR21 and LPAR22) available via PROC2.20 CTC. Use control units 5210 and 5220 and devices 5210,4 and 5220,4 respectively.
7. Close the **Edit CTC Connection** dialog to return to the **CTC Connections** dialog showing the new connection.

Create the third CTC connection

The third connection is between PROC1.20 CTC and PROC3.0.21 CNC. In this case we have a control unit and 4 devices that may be used to represent LPAR11 via PROC1.20 CTC, but we need to create a control unit and devices to represent PROC3.0 via PROC3.0.21 CNC.

1. Click **Create...** on the **CTC Connections** dialog.

2. Click the **CTC or FC CHPID** button and select PROC1.20 CTC SHR.
3. Click the second **CNC/FCV or FC CHPID** button and select PROC3.0.21 CNC SHR.
4. Click **OK**. The **Accept Proposed CTC Control Units** dialog proposes control unit 5110 to be connected to PROC3.0.21 CNC to represent LPAR11.
5. Click **Accept** to connect the control unit to PROC3.0.21.
6. The **Edit CTC Connection** dialog appears, showing that the CNC side has been defined, but there is no control unit to represent image PROC3.0 on PROC1.20 CTC.
7. Click the **Add New...** button on the left hand side to create the control unit and devices that will be connected to PROC1.20 CTC to represent PROC3.0 via PROC3.0.21 CNC. Use control unit 4300 and devices 4300,4 .
8. Close the **Edit CTC Connection** dialog. The **CTC Connections** dialog appears again showing the new connection.

Create the last CTC connection

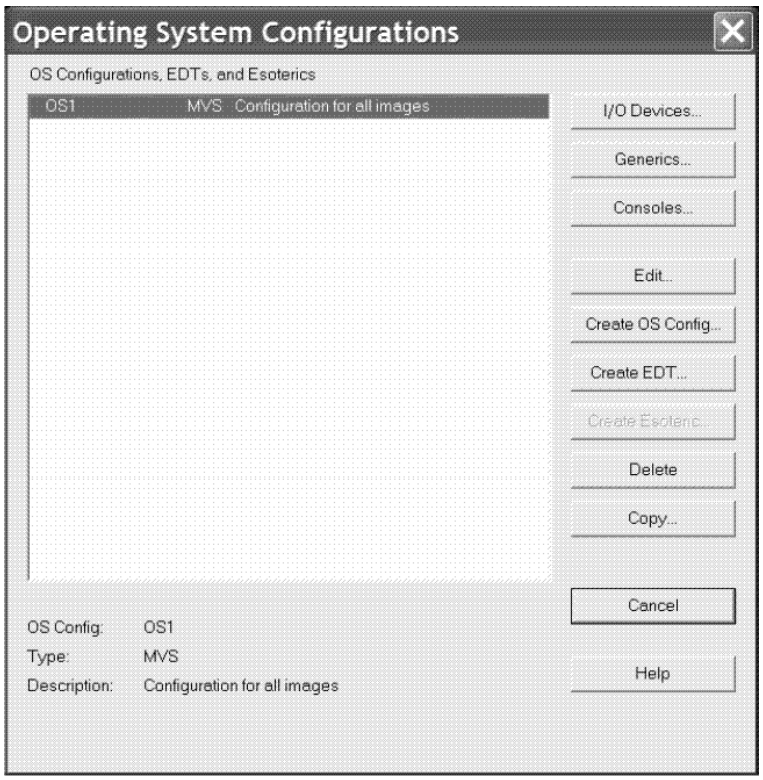
The final connection is between PROC2.20 CTC and PROC3.0.21 CNC. In this case control unit, all of the images involved are represented. During the create process, HCM proposes these control units to be connected to the CHPIDs, and we need only accept these proposals to complete the connection.

1. Click **Create...** on the **CTC Connections** dialog.
2. Click the **CTC or FC CHPID** button and select PROC2.20 CTC SHR.
3. Click the second **CNC/FCV or FC CHPID** button and select PROC3.0.21 CNC SHR.
4. Click **OK**. This invokes the **Accept Proposed CTC Control Units** dialog which proposes control unit 4300 to connected to PROC2.20 CTC to represent LPAR31 on PROC3.0, and control units 5210 and 5220 to be connected to PROC3.0.21 CNC to represent LPAR21 and LPAR22 PROC2.
5. Click **Accept** to connect the control units. The **Edit CTC Connection** dialog appears, showing the fully defined connection.
6. Close the **Edit CTC Connection** dialog. The **CTC Connections** dialog appears again, now showing all four connections. Close the**CTC Connections** dialog.

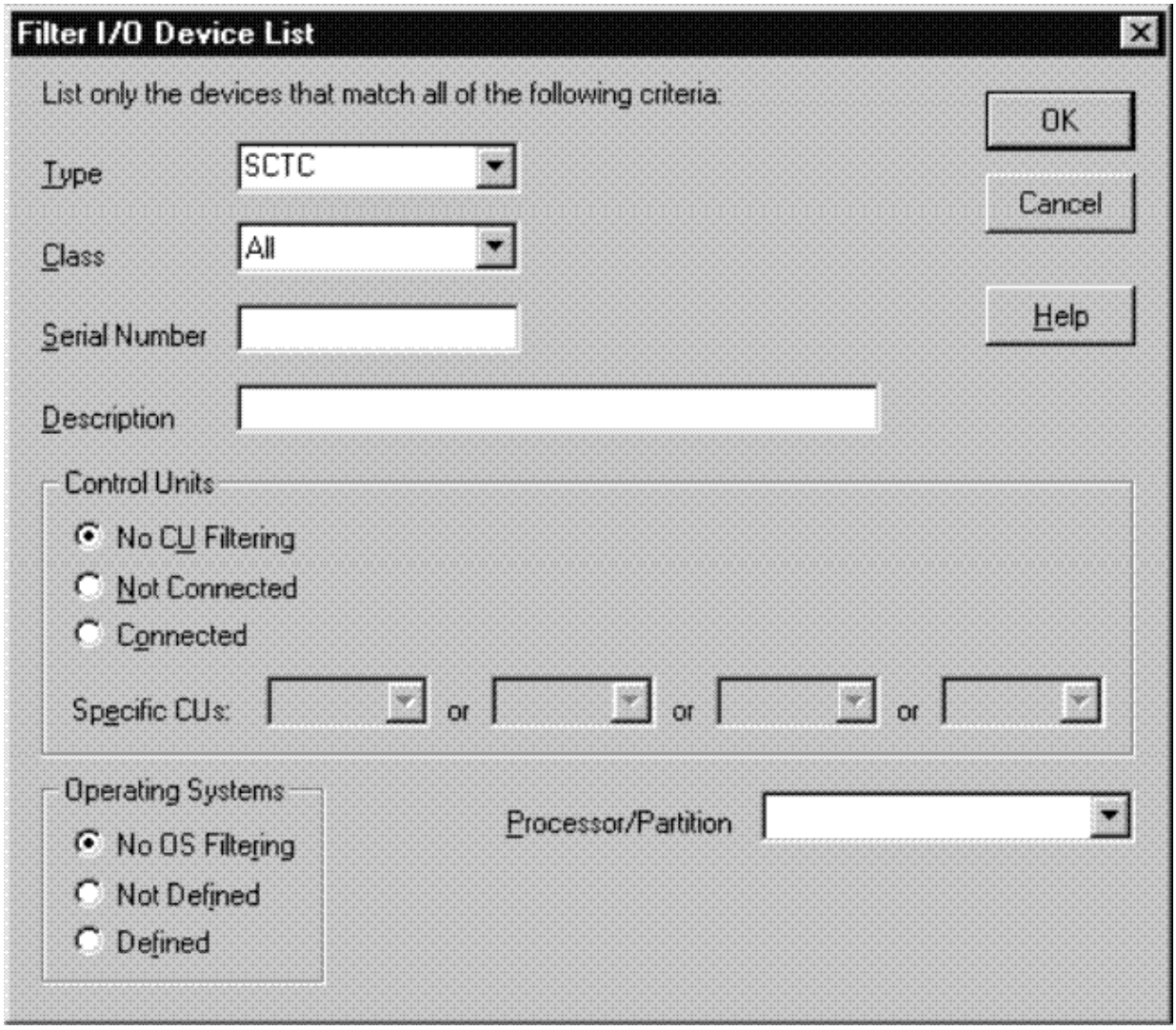
Define the CTC devices to the OS configuration

Assuming one OS configuration (OS1) is used for all four images in the connection, the last step is to define the new devices to the OS configuration.

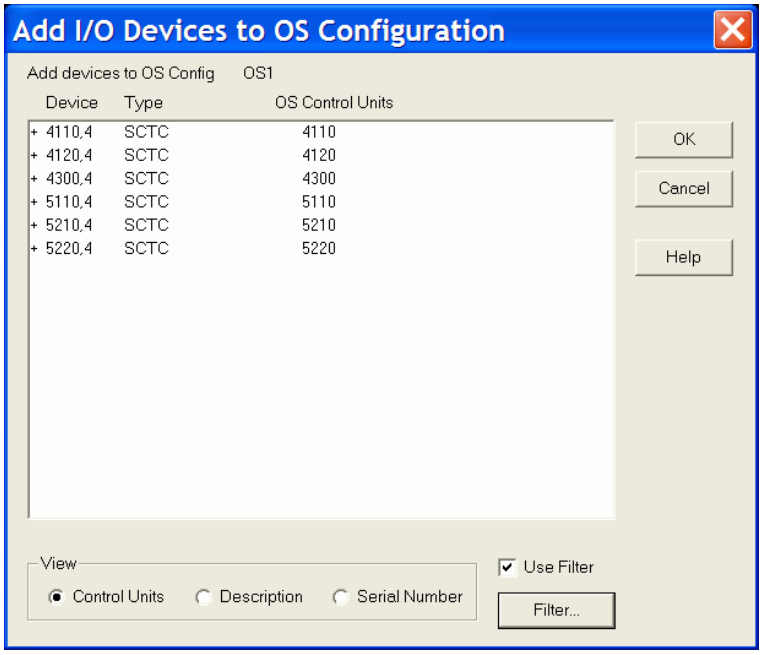
1. Select **Edit --> OS configurations...** to open the **Operating System Configurations** dialog.



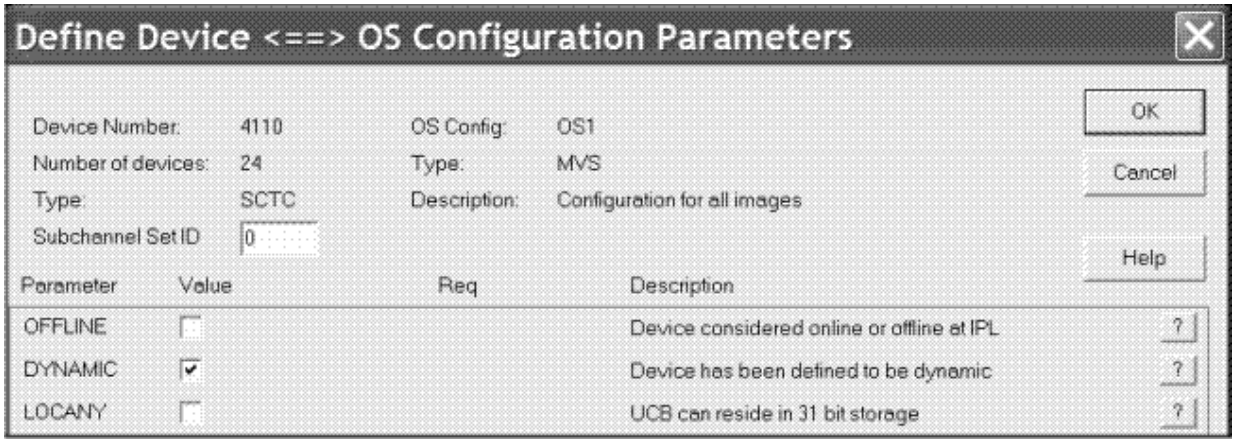
2. Select OS1 from the list box and click **I/O Devices...** to open the **List Devices for OS Configuration** dialog.
3. Click **Add...** to open the **Add I/O Devices to OS Configuration** dialog. This shows all the devices in the configuration that are not yet defined to OS1.
4. Check the **Use Filter** box and click **Filter...** to open the **Filter I/O Device List** dialog.



5. Enter “SCTC” in the **Type** field and click **OK**. The **Add I/O Devices to OS Configuration** will re appear showing only SCTC devices.



6. Select all of the devices in the list and click **OK**. The **Define Device <==> OS Configuration Parameters** dialog will appear .



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 to return to the configuration diagram.

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 nets, 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

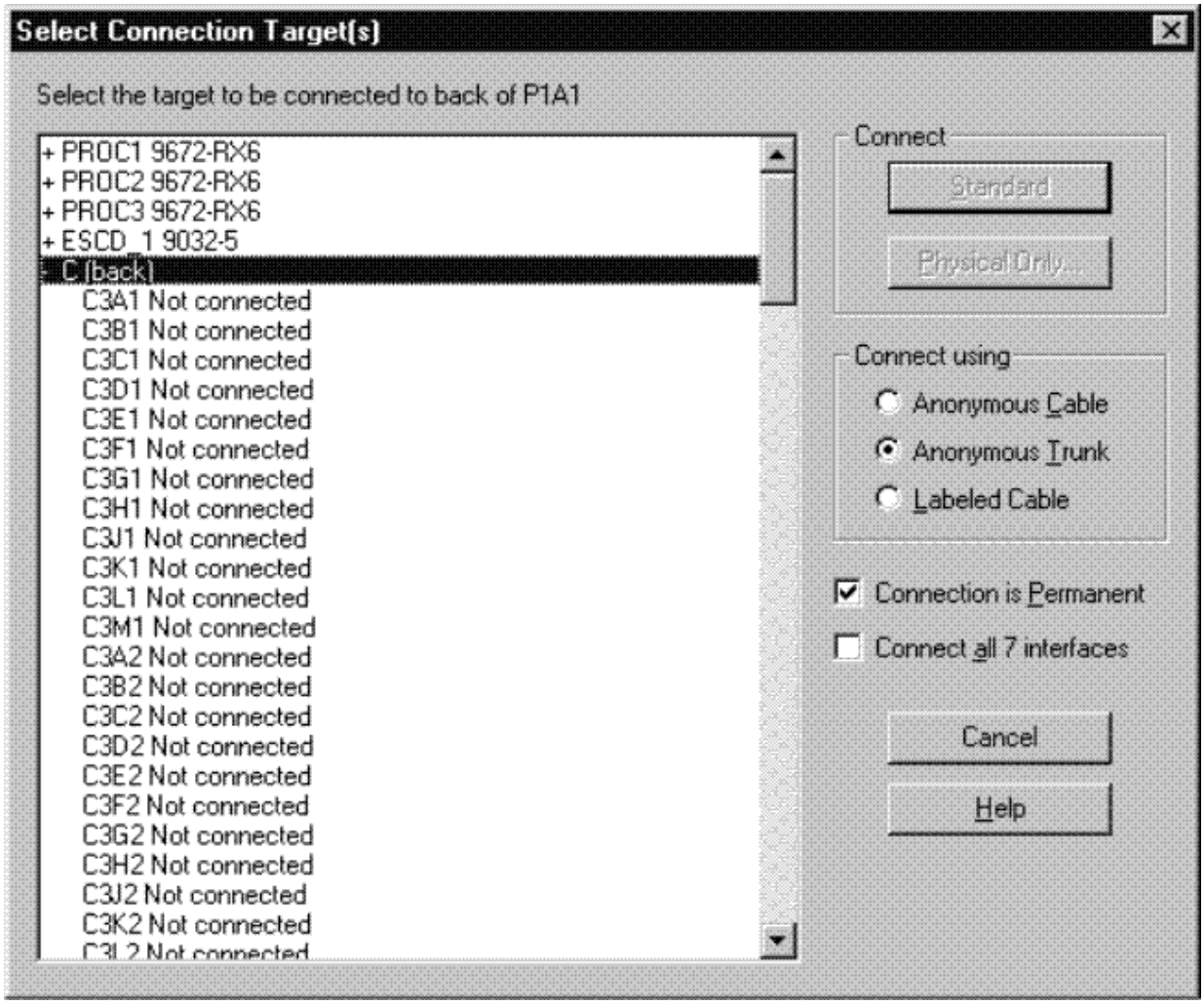
We will create the zone cabinet first.

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 the *processor zone*. Select *Near processors* as the ***Graphical Position*** for the cabinet. 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 second cabinet with *IBM Main Distribution Frame* for Model, ID *C*, and as ***Graphical Position*** select ***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. Add a panel with seven rows and columns for the requested panel.
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.



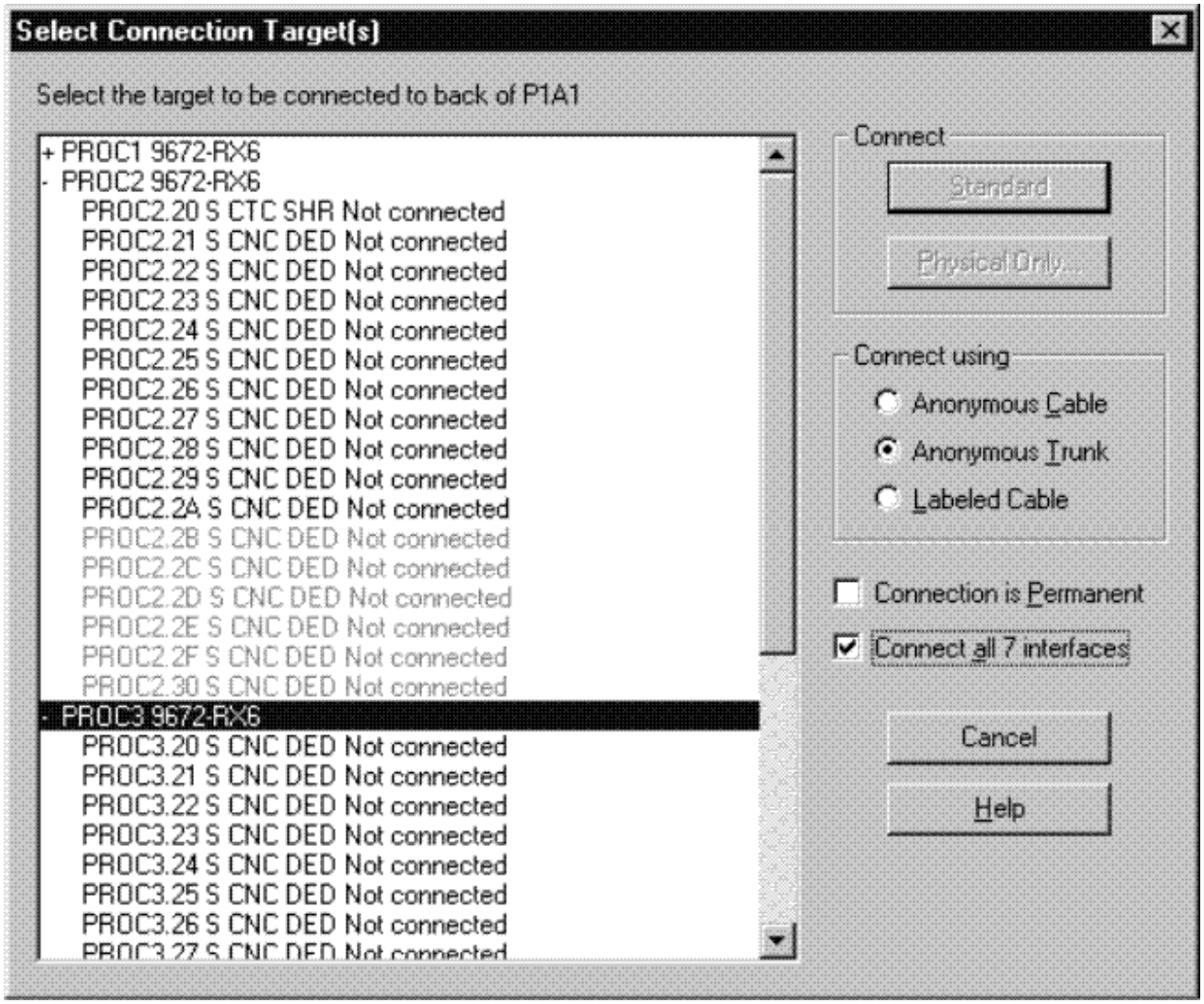
5. Select the first patchport , C3A1.
6. Check ***Connect all 7 interfaces***.
7. Select **Anonymous Trunk** and click **Physical Only...**.

   The first 7 patchports of the two cabinets have now been trunked together.

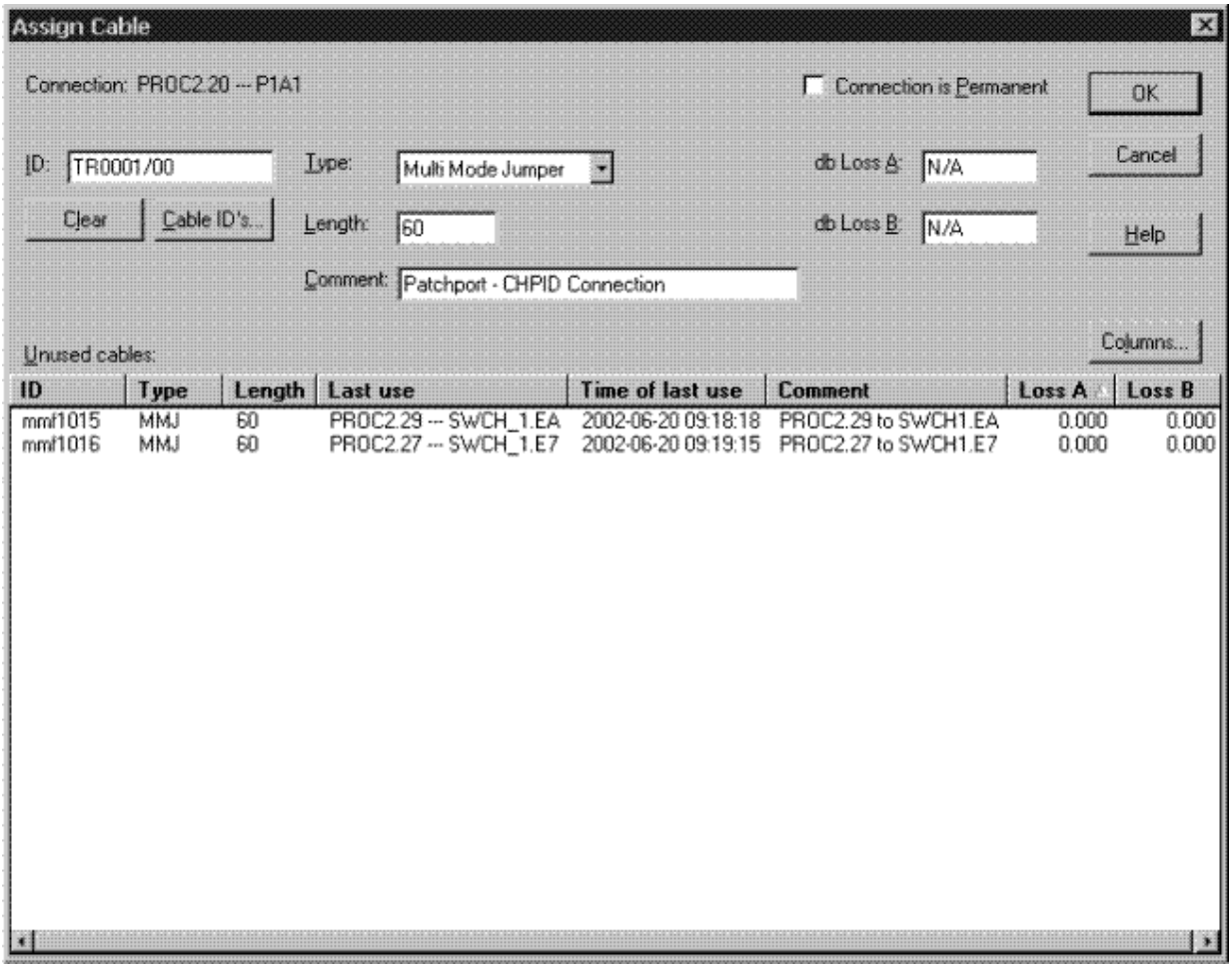
Trunk the zone cabinet to the processor

We now connect the Processor Panel to some CHPIDs on the processor, using an anonymous (unlabeled) permanent trunk.

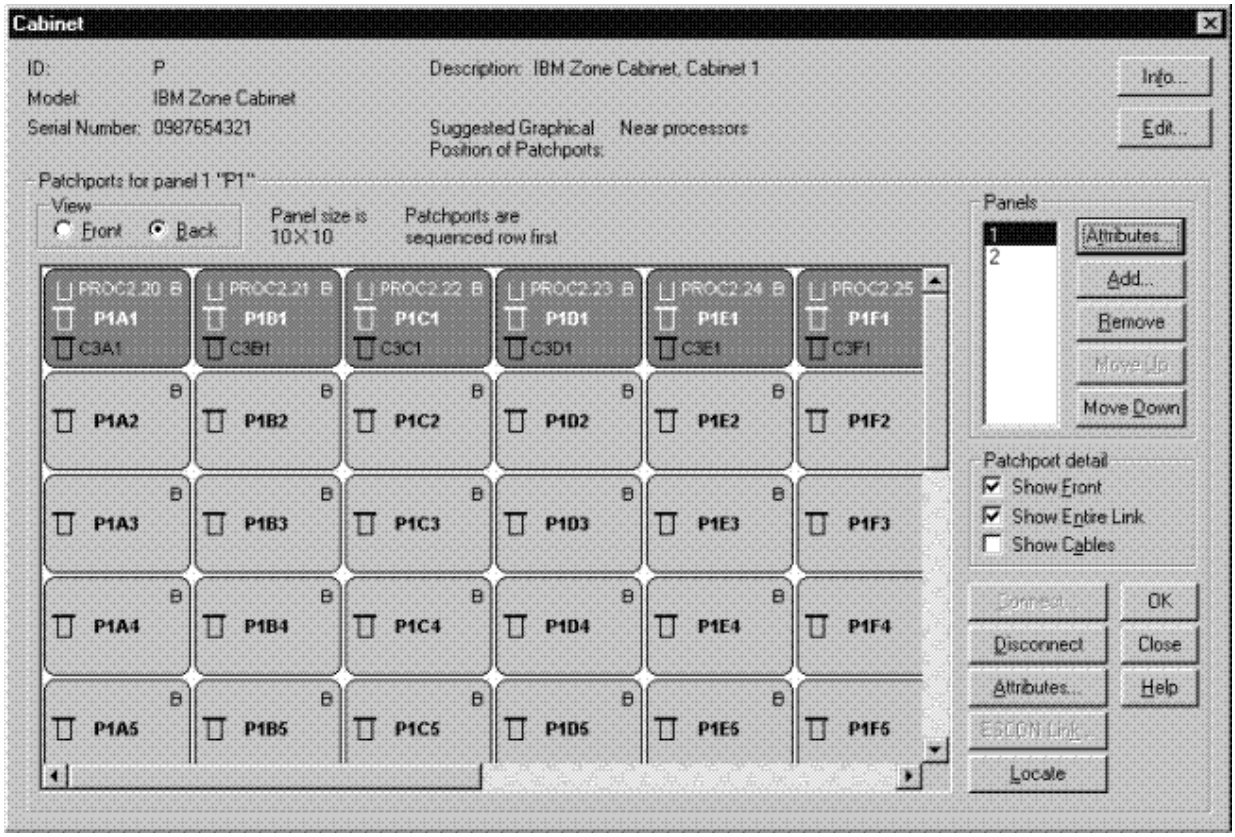
Go to the **Cabinet** dialog. Change the radio button from ***Front*** to ***Back*** in the View section and click ***Connect***. HCM opens the **Select Connection Target(s)** dialog.



1. In the list box, double-click 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**. HCM opens the **Assign Cable** dialog six times in sequence for you to assign labels to the individual lines of the trunk.



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.



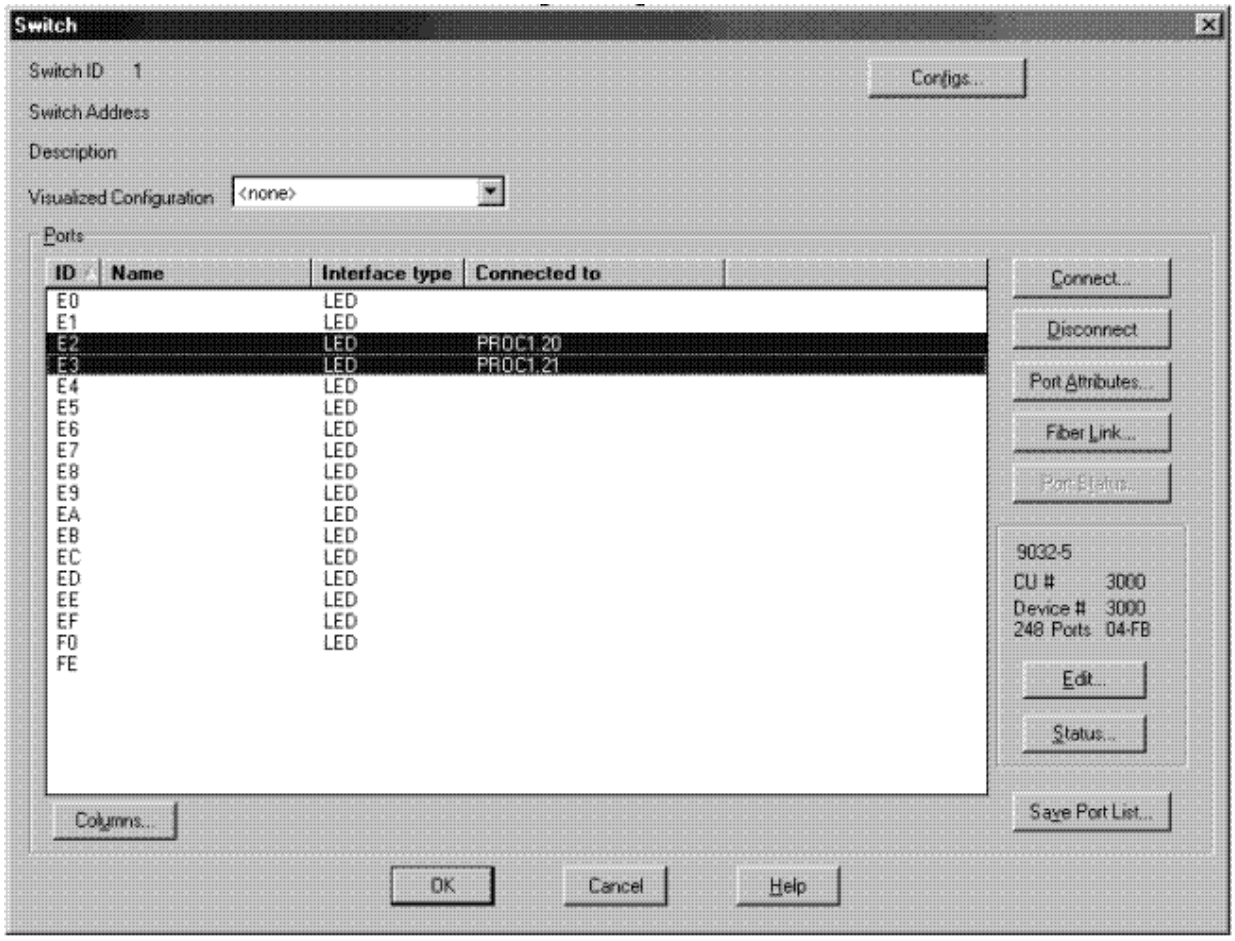
7. Click **OK** on all open dialogs to close them.

Connect a switch to CHPIDs through the patchport cabinets

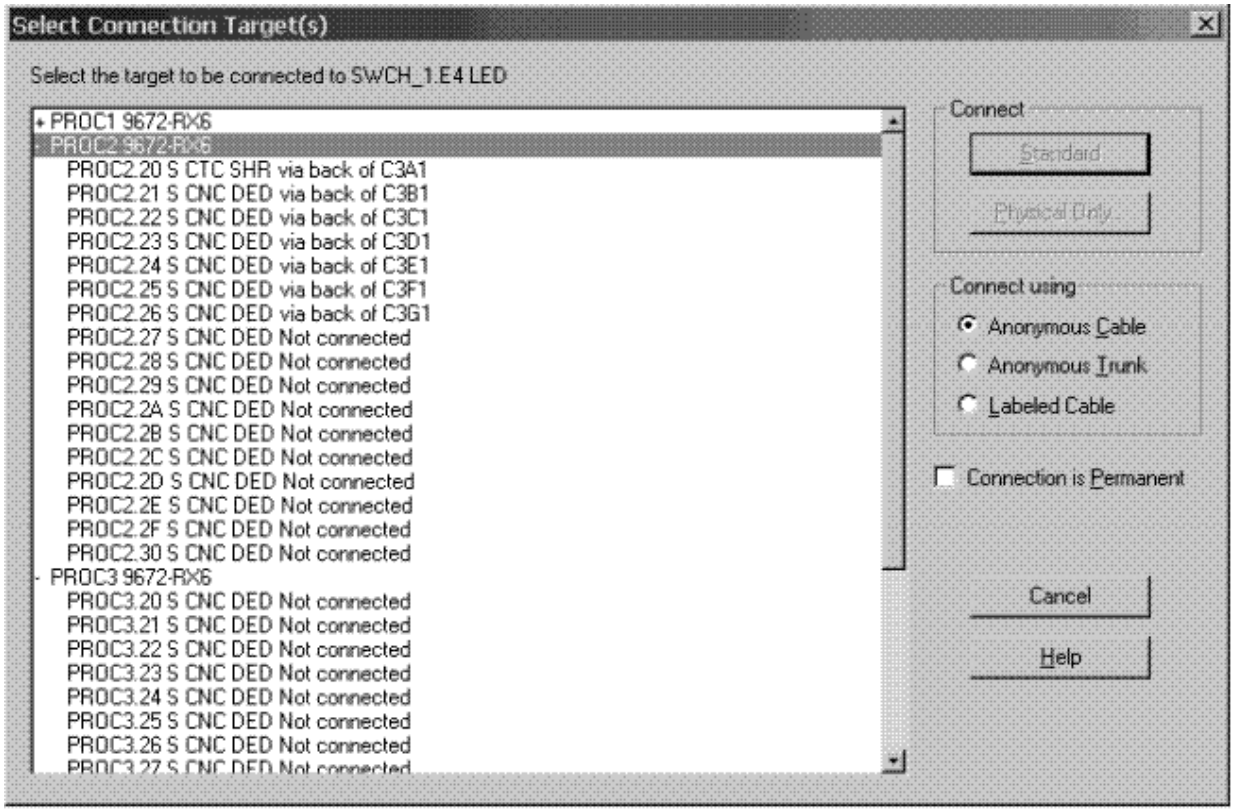
Permanent connections between pairs of patchports must be made before permanent connections between patchports and CHPIDs. It is also necessary to make all permanent (trunk) connections among patchports before adding those patchports to fiber links.

We 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.



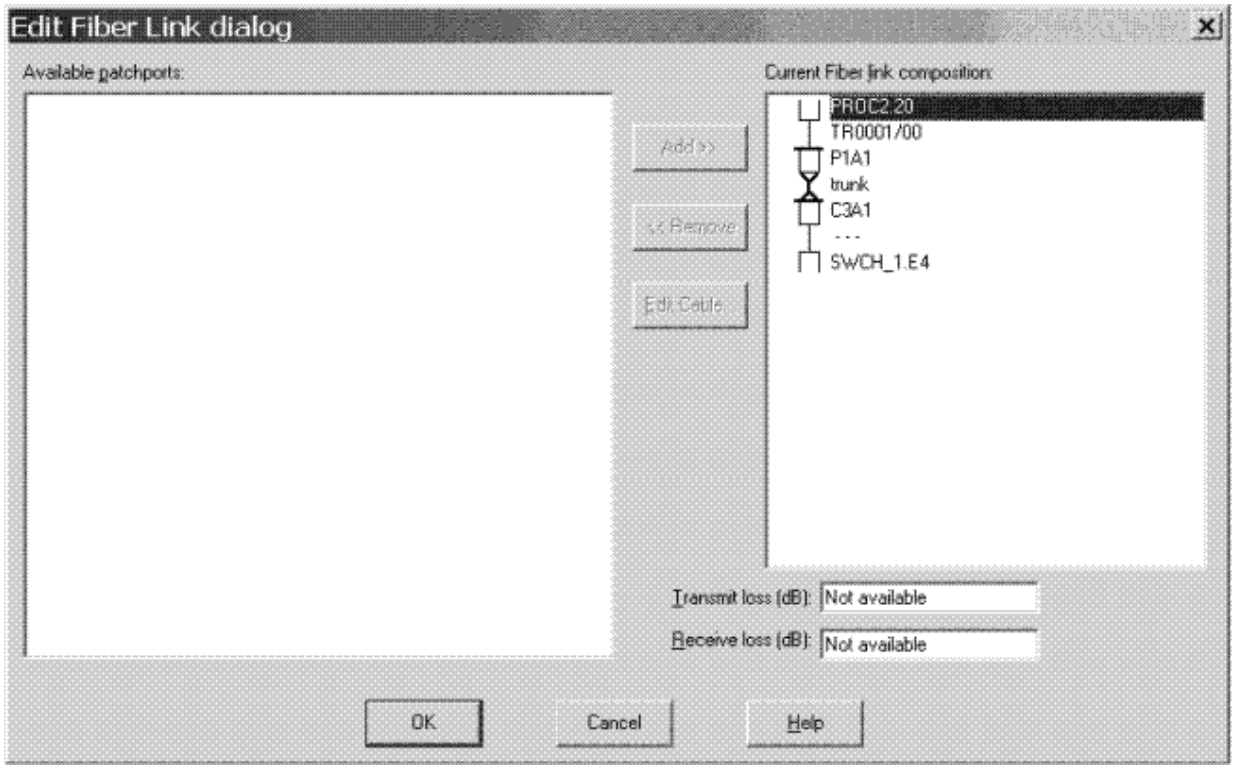
2. Select one of the free switch ports and click **Connect**. This opens the **Select Connection Target(s)** dialog.



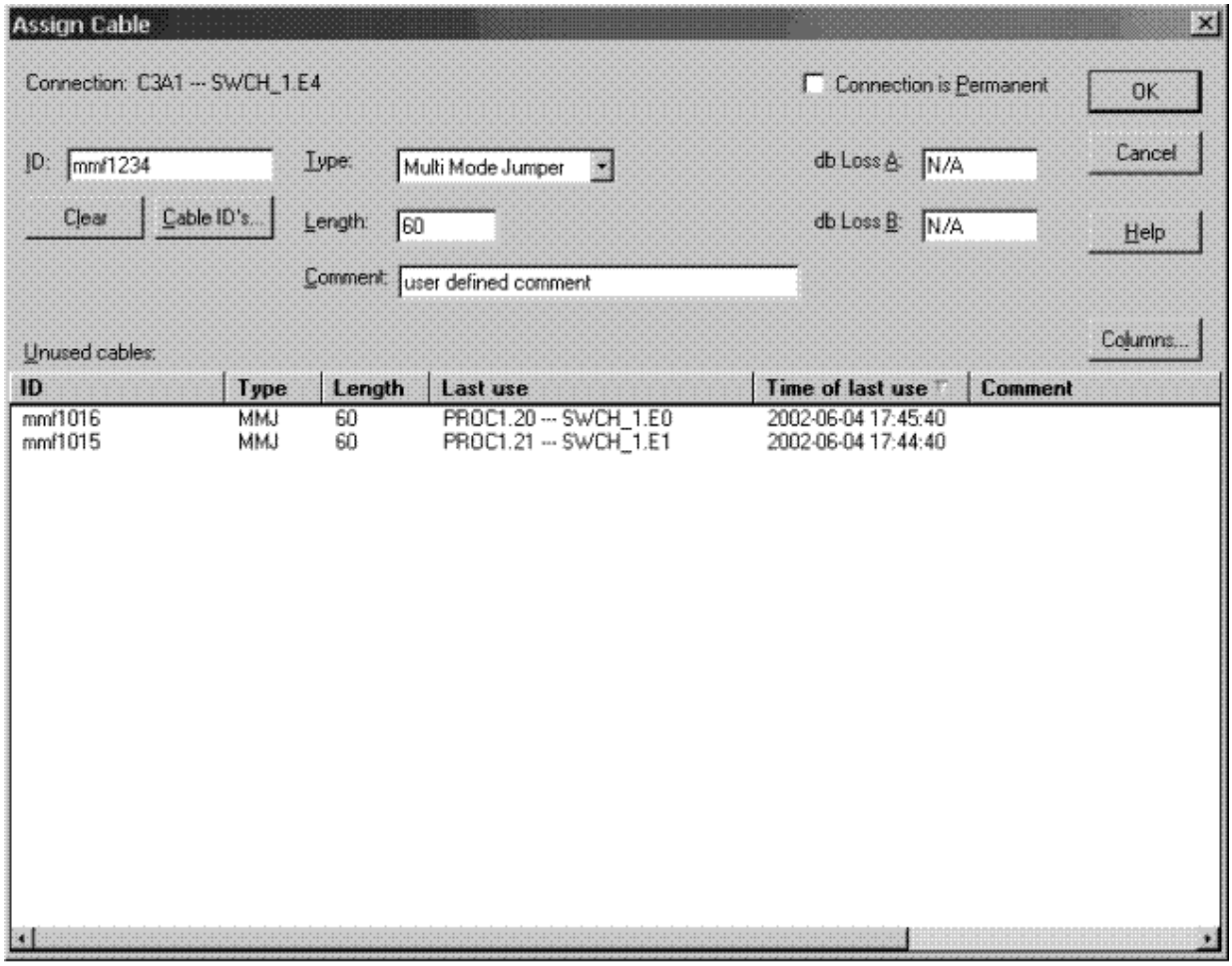
3. In the list box, double - click on the processor with the trunked CHPIDs. This expands the list, showing all CHPIDs in the processor.
4. Double - click on one of the trunked CHPIDs. The connection is made between the switch port and the CHPID.

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.



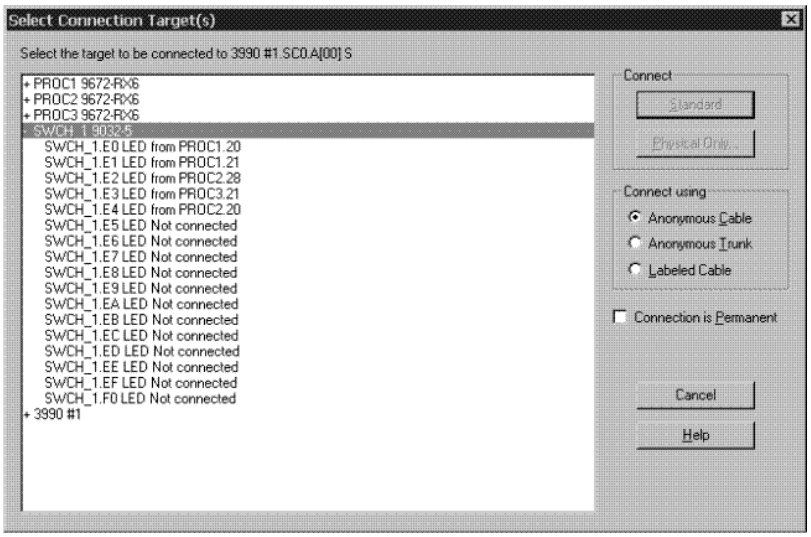
2. Select the “--” between the switch port and patchport C3A1 and click **Edit Cable**. This opens the **Assign Cable** dialog.



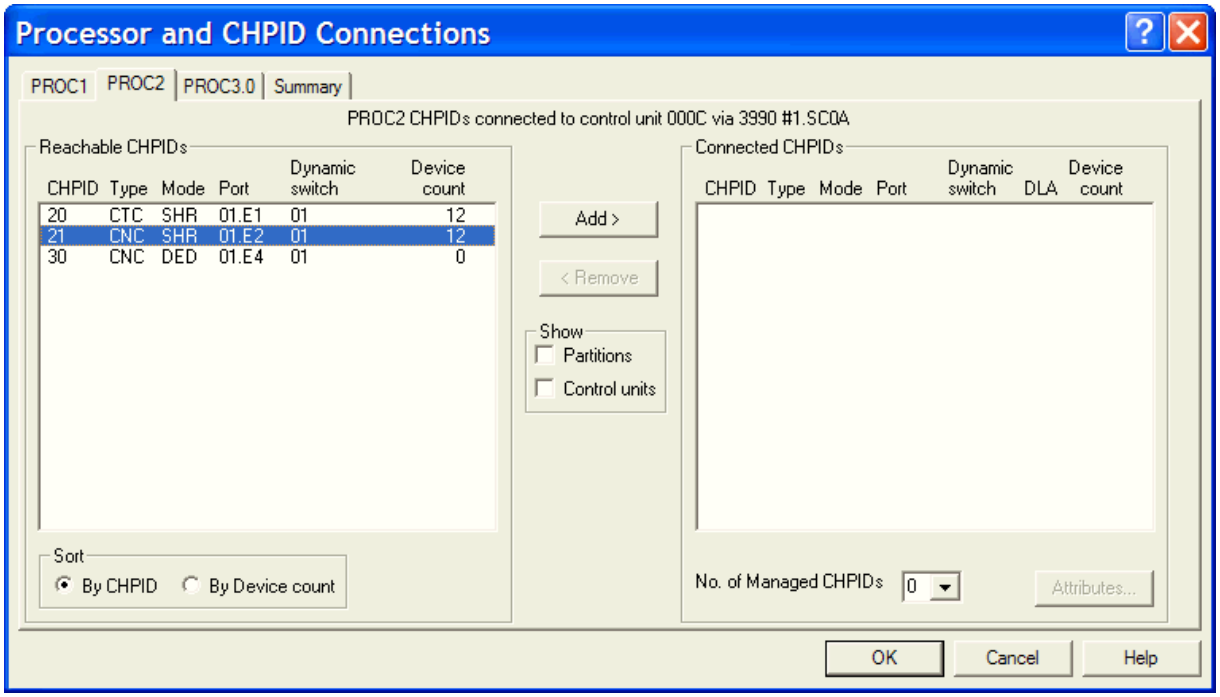
3. Fill nt BackColor="#FFC0FF">in the cable 's ID, type, and length, then click **OK**. The cable 's new label now appears in the **Fiber link** dialog.
4. Click **OK** on all open dialogs to close them. The new connection between the switch and the CHPID through both patchports can be seen on the diagram.

Connect a controller to the CHPID via the switch

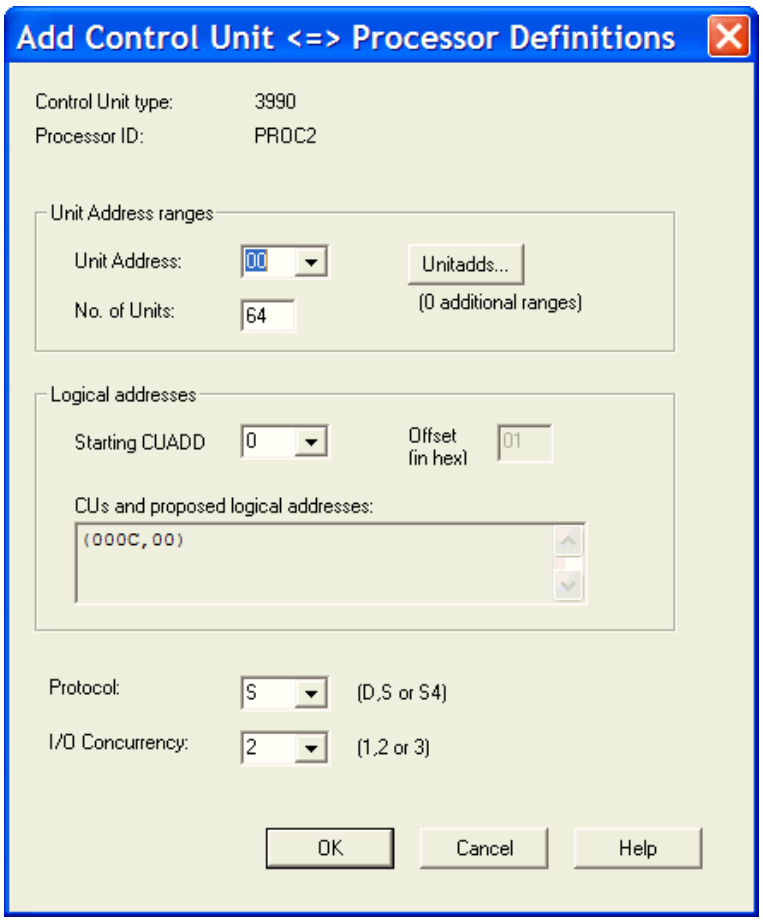
1. Edit an existing controller by selecting **Edit --> Controller** from the menu. This opens the **Edit a controller...** dialog. Select the controller that you want to connect from the list and click **OK**. This opens the **Controller** dialog. As an alternative, you can directly look for and double-click on the desired controller in the diagram to reach the **Controller** dialog.
2. Select a free channel interface ***(Not connected)*** from the ***Interfaces*** section in the dialog and click **Connect**. This opens the **Select Connection Target(s)** dialog.



3. Choose the switch that we trunked to a CHPID from the list and double-click it to expand it. This lists all 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.



6. Select ***PROC2*** from the shown tabs on the top of the dialog and select CHPID 21 CNC that we trunked to this switch and click **Add>>** to move this CHPID to the **Connected CHPIDs** list. This opens the ***Add Control Unit <=> Processor Definitions*** dialog.
7. In the ***Add Control Unit <=> Processor Definitions*** dialog, select the desired connection properties and click **OK**.



8. HCM returns to the **Processor and CHPID Connections** dialog where CHPID 21 CNC moved to the **Connected CHPIDs** list. Click **OK**. The connection is now made and HCM returns to the **Controller** dialog.
9. 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.
10. 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:

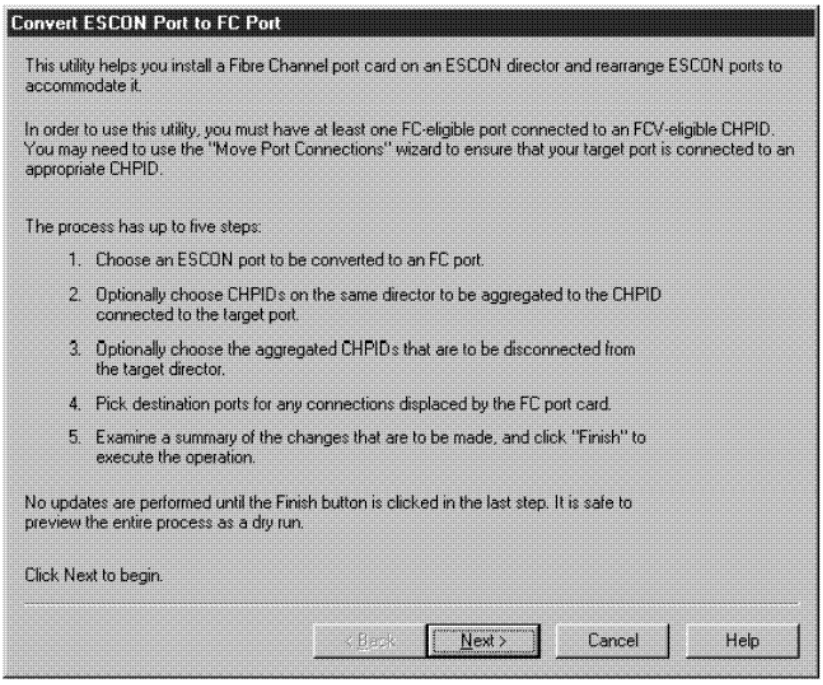
- FCV channels can only be installed in specific locations in the processor and have particular CHPID numbers depending on the processor type.
- FICON bridge cards can have only one port address (and link address), on a modulo 8 + 4 port address.
- The 7 consecutive subsequent port addresses of a FICON 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

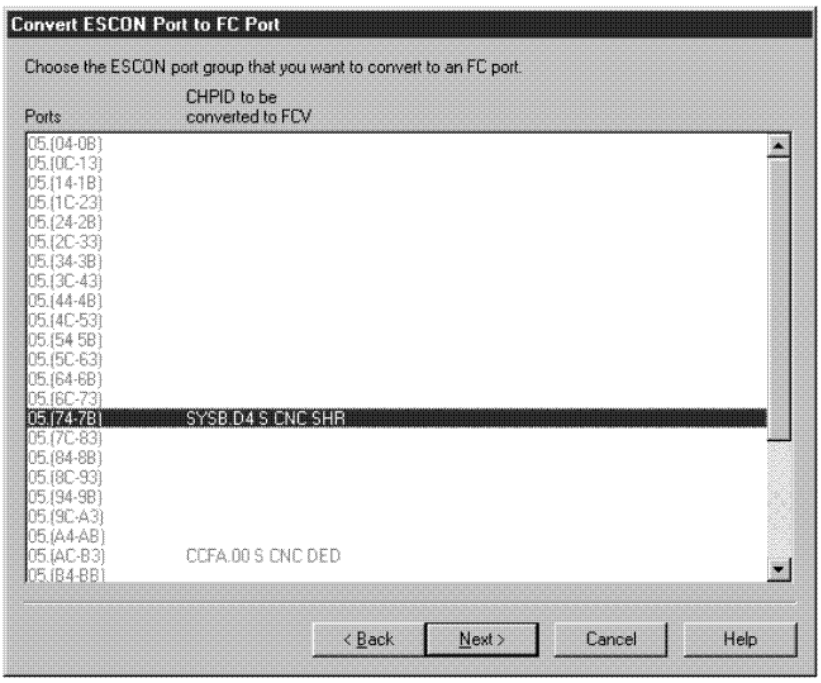
D4 must be converted from CHPID type CNC to FCV. Also, all attached I/O equipment of CHPID D7 is disconnected and reconnected to the FCV CHPID D4 (aggregation of CHPID D7 to D4) to make use of the wider bandwidth of the FCV channel. The original connection between ESCD port E0 and CHPID D7 will be disconnected (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 . Re locating 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.

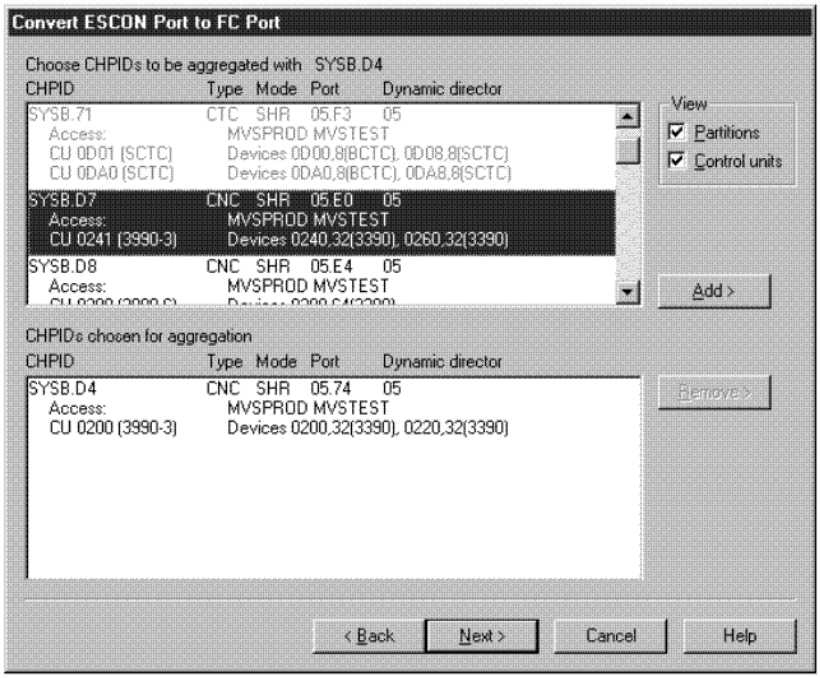


   Click the **Next >** button to continue.
3. A list of all possible ports that can be used to hold a FICON bridge card will be shown . The only selectable ports will be those with an appropriate port address (starting with 04 modulo 8 + 4 of 9032-5) and which are already connected to an FCV or to an FCV-eligible CHPID (for example, a CHPID whose type can be changed to FCV). All ports that are not usable for a FICON bridge card are grayed out .



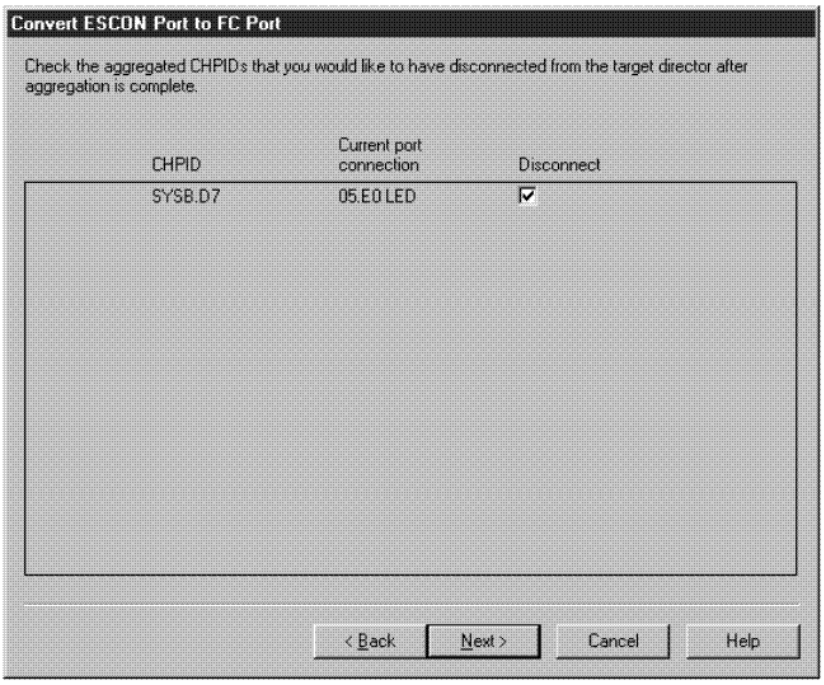
Select port 74 of director 05 connected to CHPID D4 of processor SYSB and click **Next >**.

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.



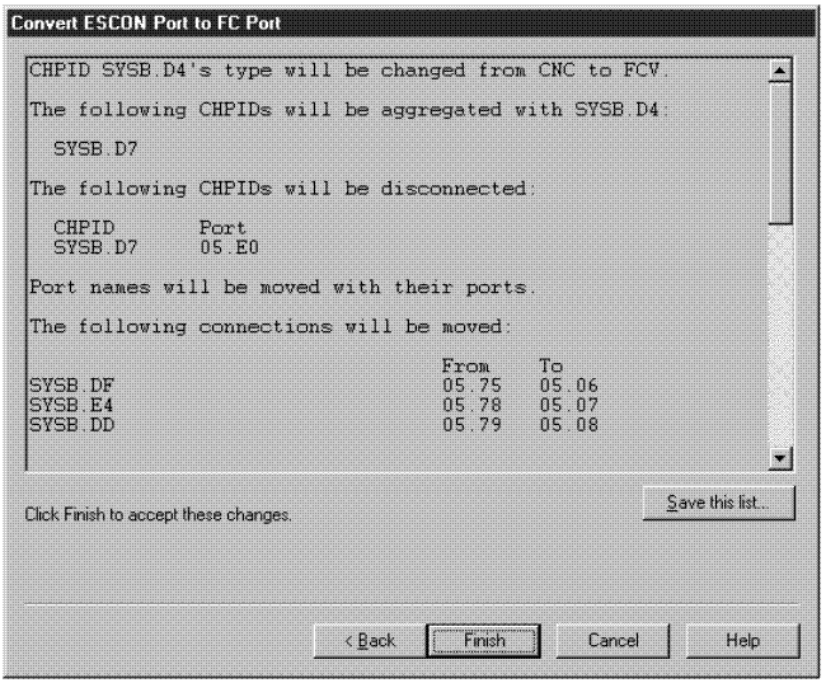
In the bottom part of the window you see CHPIDs and their I/O equipment as it is aggregated. Click **Next >** to continue.

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 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 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 beont BackColor="#7DFB87">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 un installed :

```
  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 guides 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.

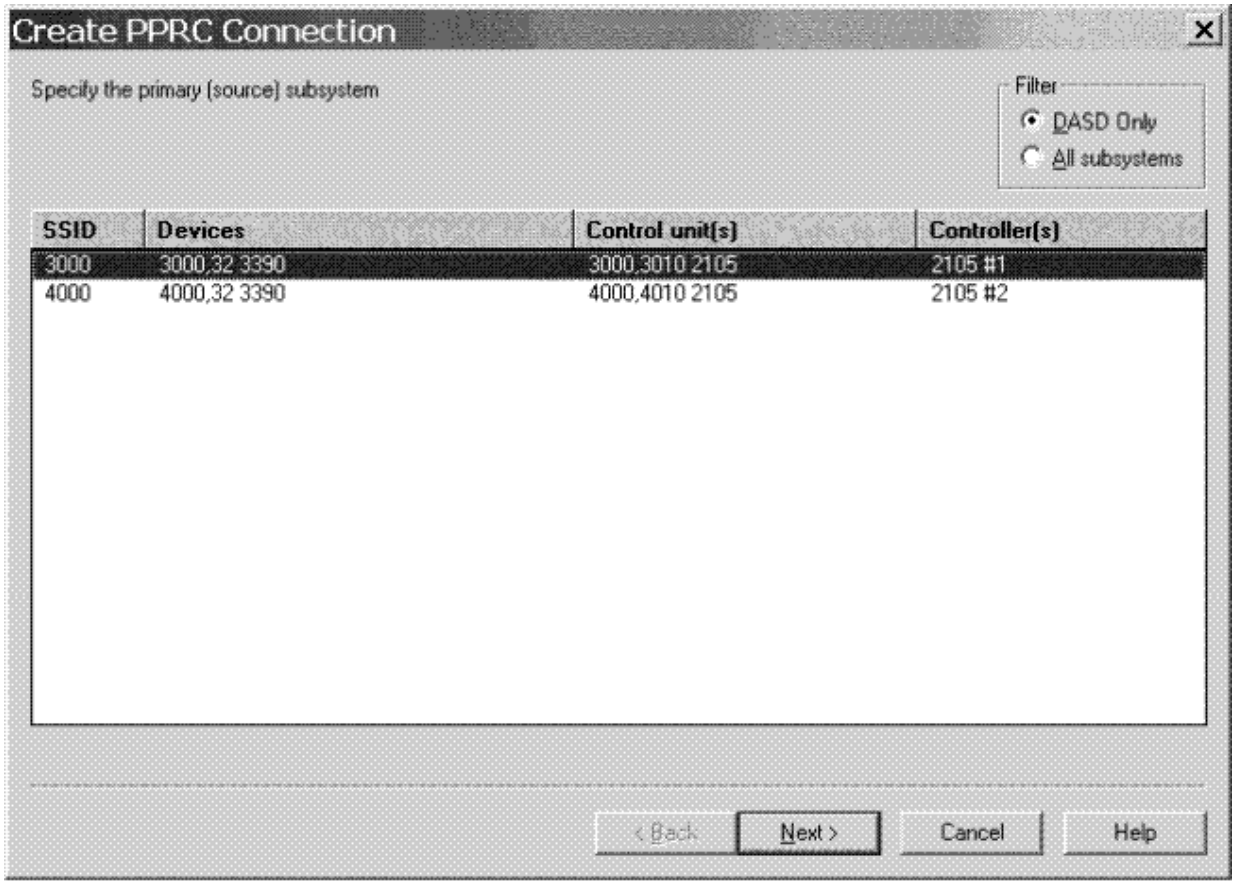


Figure 184. Create PPRC Connection (1)

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 automatically fills the entries on the right side (target interfaces). Your **Create PPRC Connection** dialog should now look similar to the following:

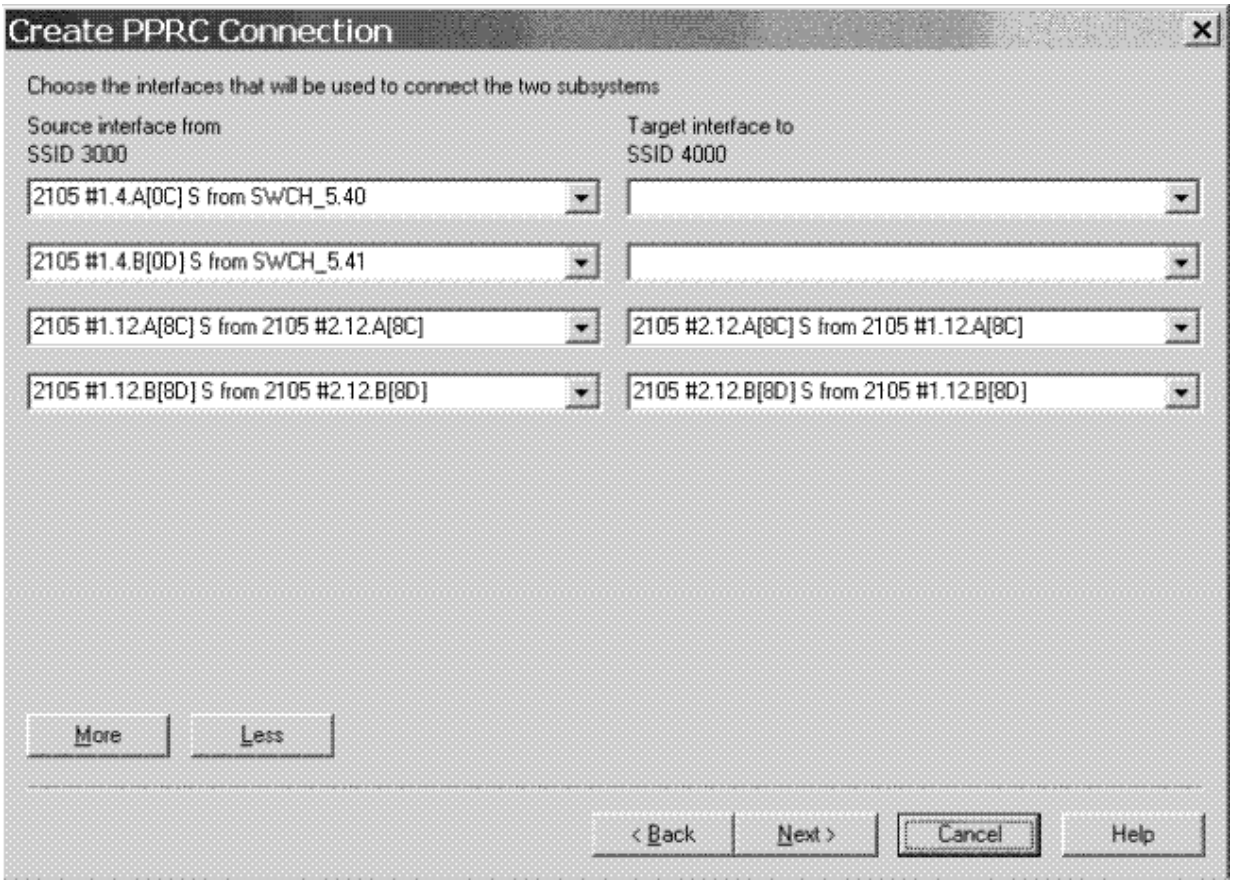


Figure 185. Create PPRC Connection (2)

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.

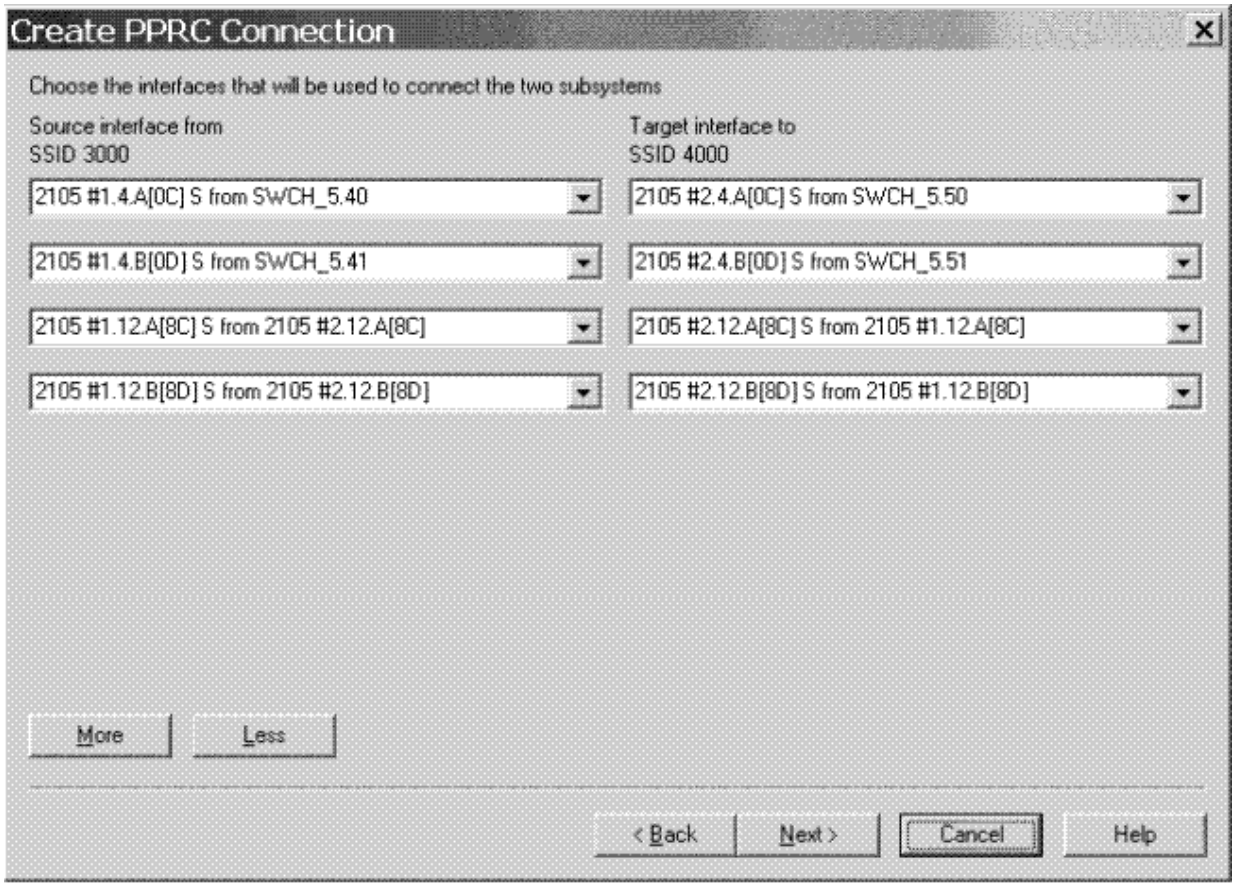


Figure 186. Create PPRC Connection (3)

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.

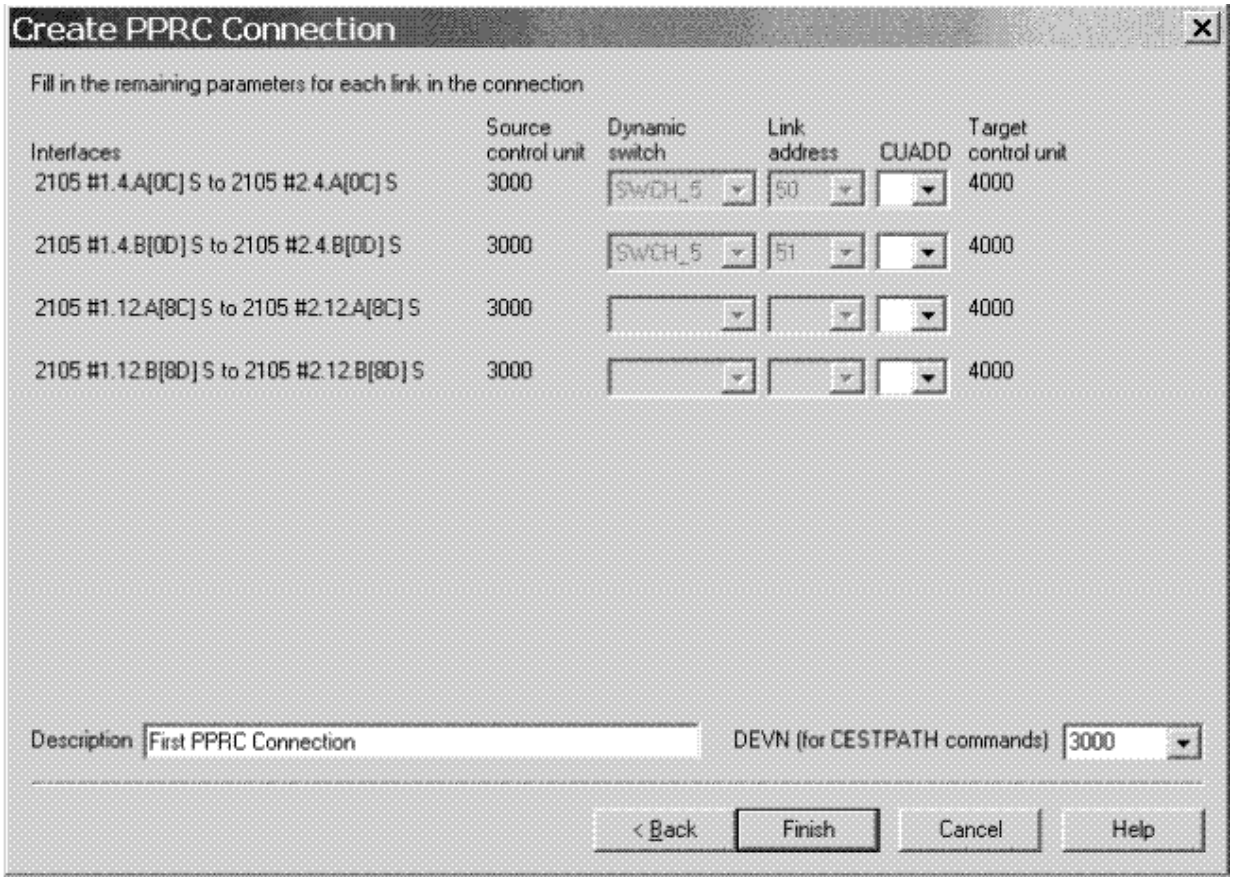


Figure 187. Create PPRC Connection (4)

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 13. 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*.

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*.

Upgrade an IODF from an earlier version of HCD

z/OS V1.7 HCD starts using the new V5 IODF format. If you want to use a lower version IODF with z/OS 1.7 HCD or higher, you first have to upgrade the IODF into the new format before using it.

You can also perform this task using HCM: see "Upgrading the associated IODF from version 4 to version 5 format" on page 70.

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

Only on HCD, configuration packages allow configuration changes made at a remote system to be merged back into the master IODF.

Note: The support of configuration packages via a dialog is not available with z/VM HCD.

© Copyright IBM Corp. 1995, 2021

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)

For more information, refer to Build an IOCDS in *z/OS HCD User's Guide*.

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

For more information, refer to Activate a configuration dynamically in *z/OS HCD User's Guide*.

Build configuration data sets

With HCD you can build configuration data sets for JES3 initialization stream checker and HCPRIO.

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 14. 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 about the data format, see “Data format for importing and exporting” on page 333. For a list of tables that are available in HCM, see “Table summary” on page 347.

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.

Note:

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” on page 334.

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

© Copyright IBM Corp. 1995, 2021

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 PC 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 347 and “Detailed table formats” on page 349.

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 quotation marks (the usual quotation marks used in prose, for example, "A, and B"), or in single quotation marks (the apostrophe character, for example, 'C, and D'). Use "" within a double-quoted string to embed the double quotation marks - "He said ""Hello""." . Single quotation marks can be used similarly, when single quotation marks are used to enclose the entire string. The Export function encloses text fields in double quotation marks.

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 quotation marks 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 quotation marks 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 quotation marks 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 quotation marks 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 quotation marks 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 CONNECT Table). 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 about the data formats, table names, and table details, refer to “Data format for importing and exporting” on page 333, “Table summary” on page 347, and “Detailed table formats” on page 349.

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 333.

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.

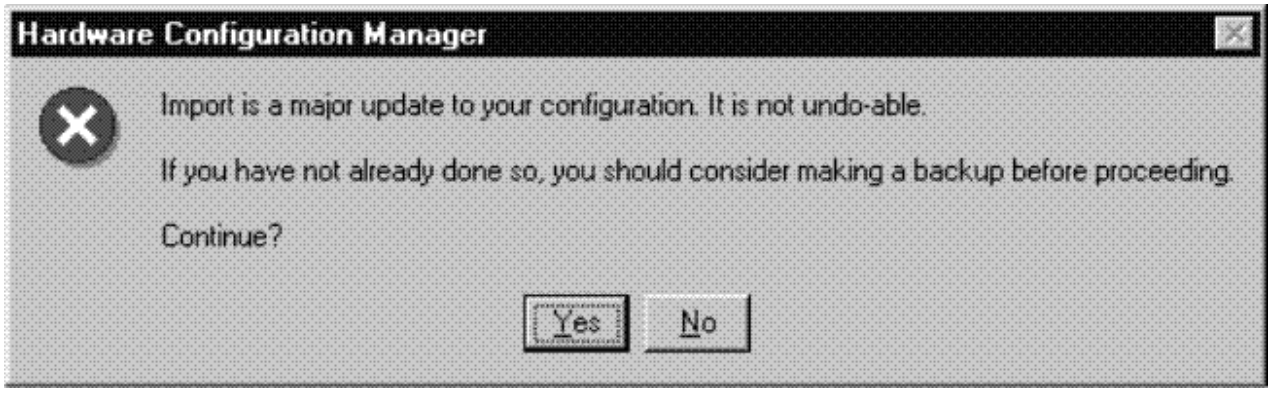


Figure 188. Hardware Configuration Manager Confirmation Message

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 189. 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**.

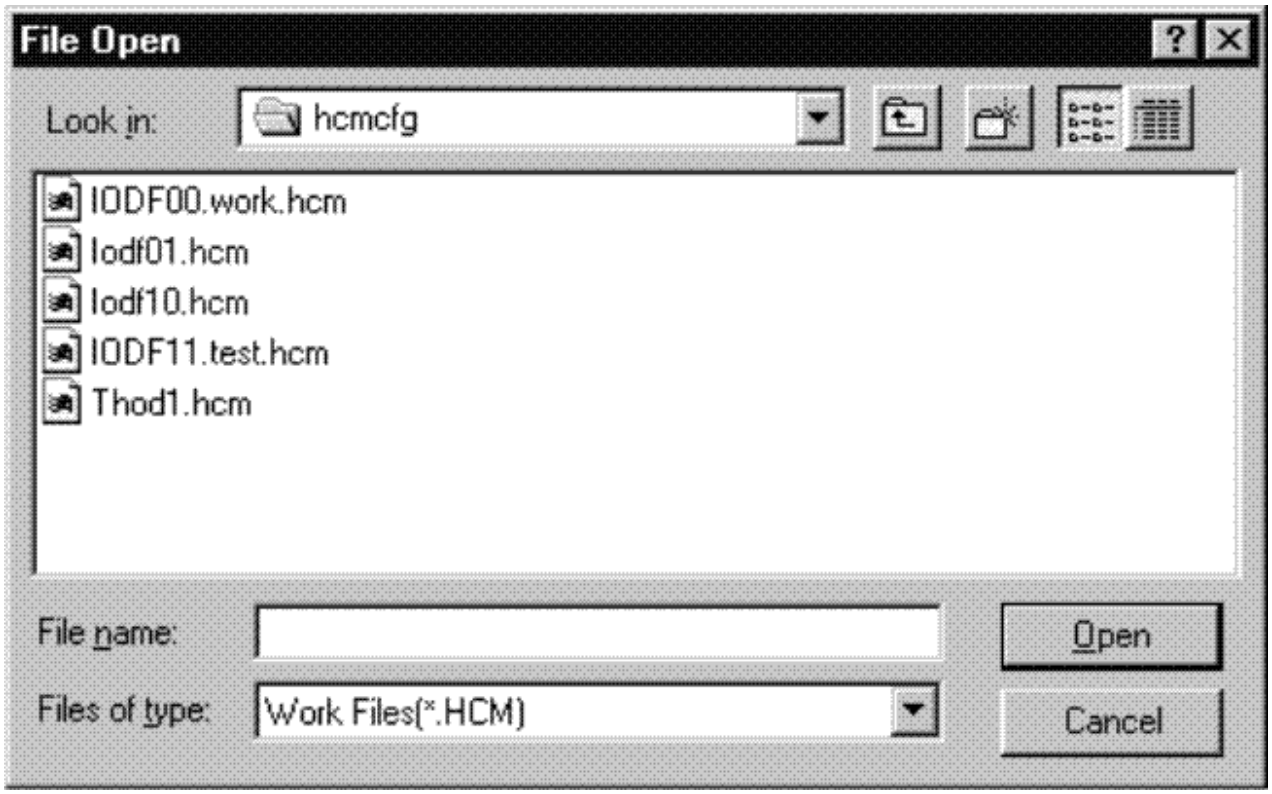


Figure 190. File Open Dialog

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.

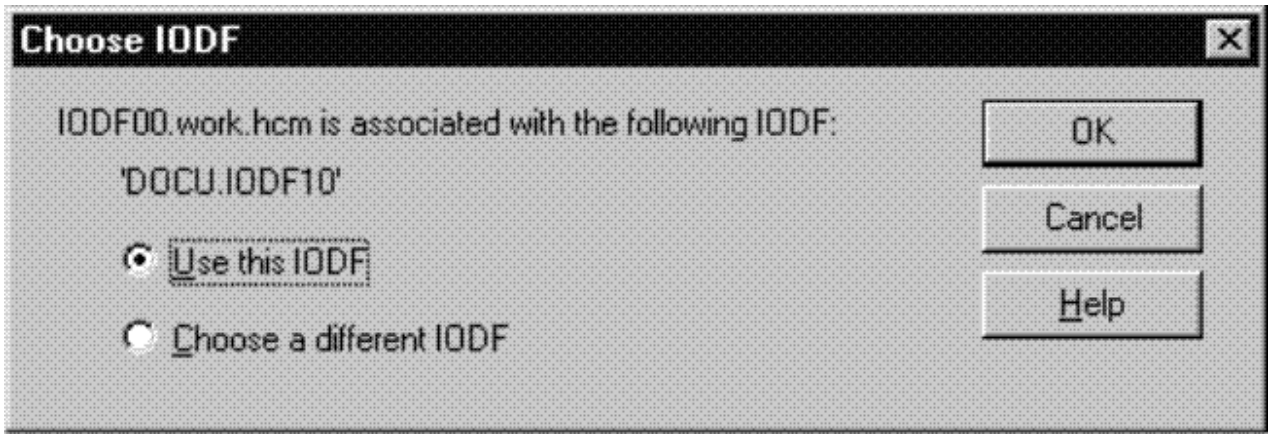


Figure 191. Choose IODF Dialog

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.)

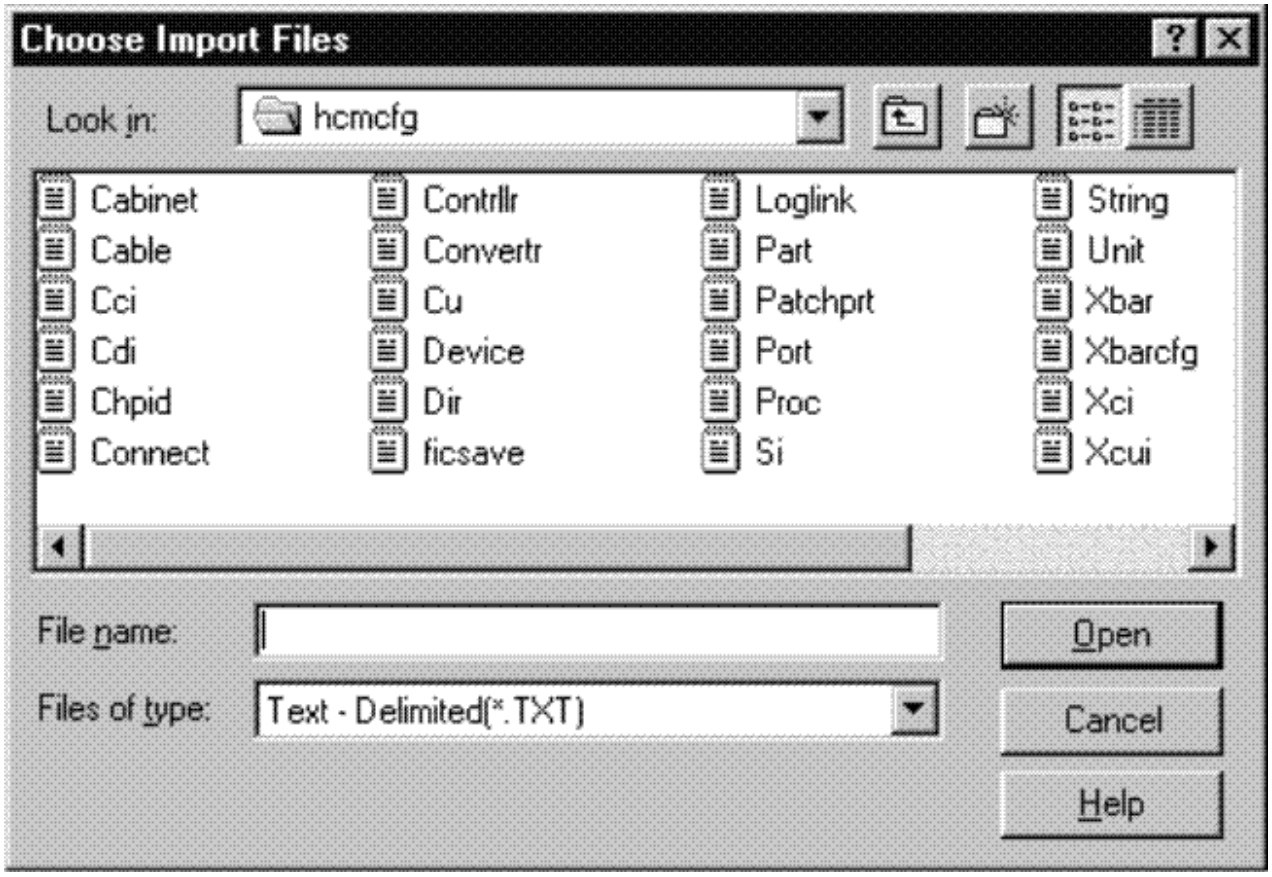


Figure 192. Choose Import Files Dialog

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.

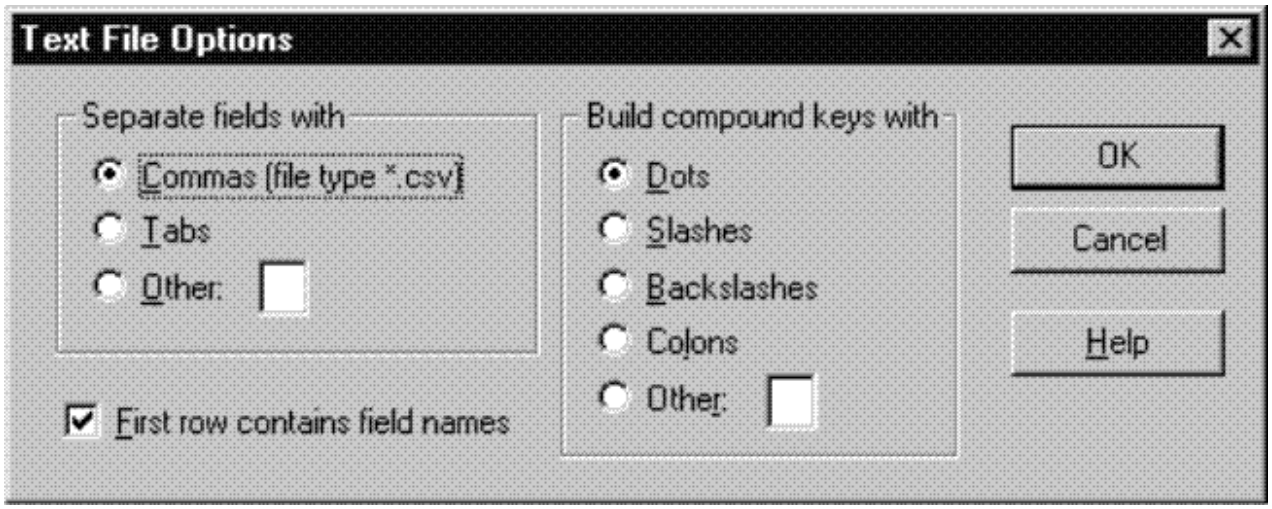


Figure 193. Text File Options dialog

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.

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.

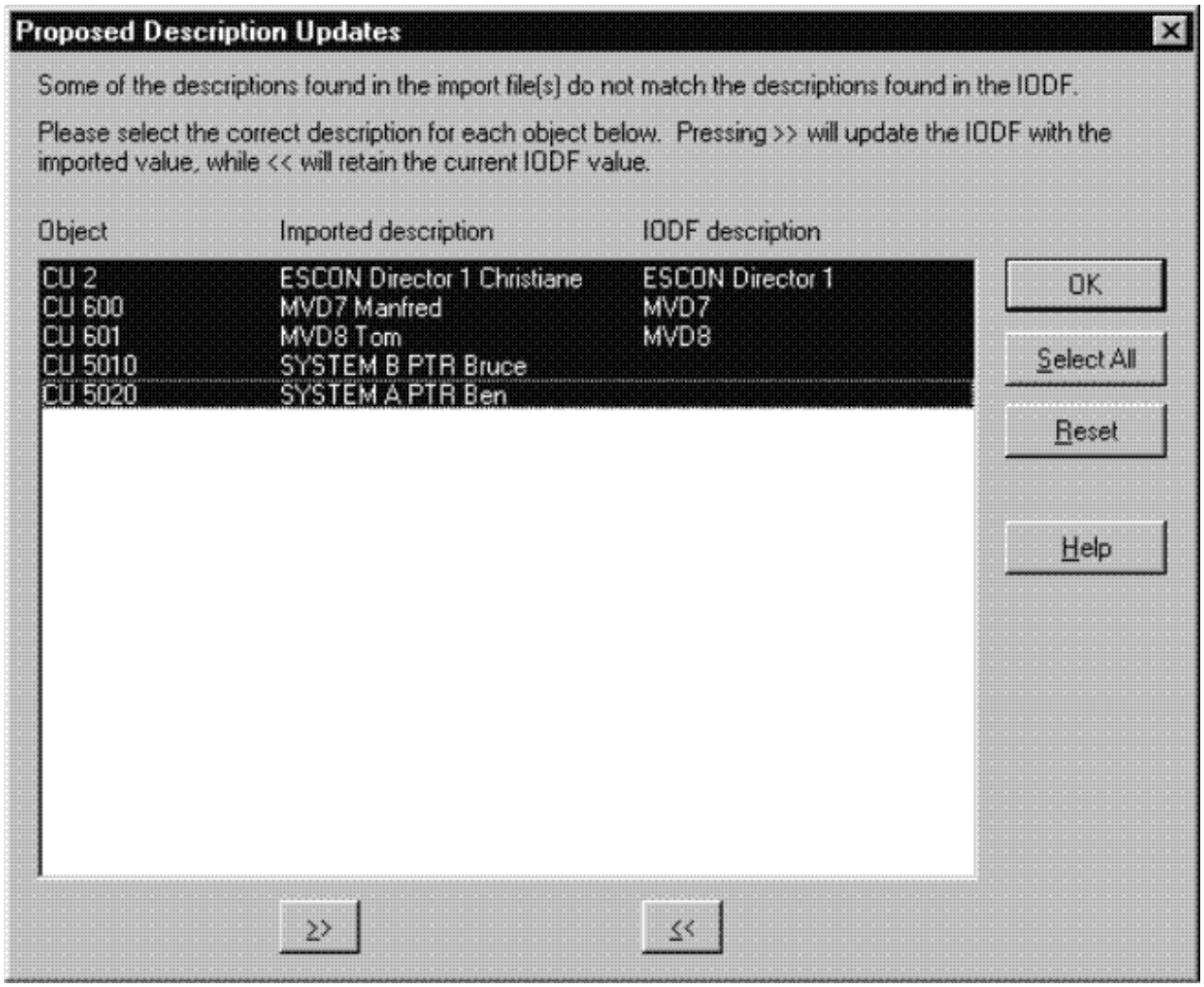


Figure 194. Proposed Description Updates Dialog

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 are 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” on page 343 above is applied again, then step “2” on page 343, 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 an 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.

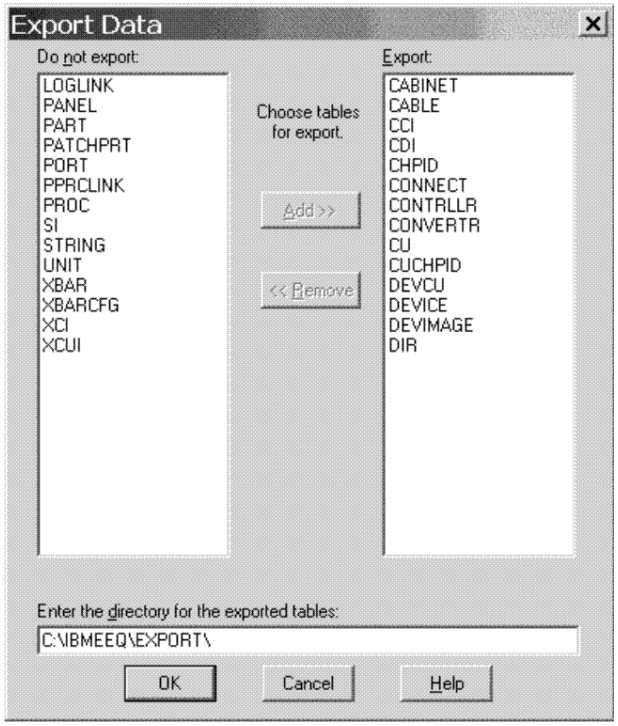


Figure 195. Export Data Dialog

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 (see Figure 193 on page 341).

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 online. For more information see “VOLSER reconciliation” on page 97.

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 and general boxes that you wish to create need to appear in the import file.

PANEL
All panels you wish to create within a specific cabinet or general box.

PATCHPRT
Entries in this table need only be provided when updating or specifying attributes for these objects. Patchports or general box ports 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 345 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.

Note: In HCM, by default the table fields are separated by a comma (",") and compound keys or fields are connected by a dot ("."). You can change these defaults during the **Export Data** dialog. The description of the tables and examples in this chapter assume a slash ("/"). See also “Record structure” on page 334.

Table 13. Table summary

Category	Table name	Description
Processor related	**PROC**	a processor, see “PROC table” on page 349
	PART	a partition, see “PART table” on page 351
	CHPID	a CHPID, see “CHPID table” on page 352
Switch related	**DIR**	a switch, see “DIR table” on page 354
	PORT	a switch port, see “PORT table” on page 354
Controller/Control unit related	**CONTRLLR**	a physical controller, see “CONTRLLR table” on page 355
	CCI	a controller channel interface, the interfaces on the top of a controller: A, B, C ..., see “Controller channel interface (CCI) table” on page 357
	CDI	a controller device interface, the interfaces on the bottom of the controller: SP0, SP1, ..., see “Controller device interface (CDI) table” on page 358
	CU	an IODF control unit, see “CU table” on page 359
String/Device related	**STRING**	a physical string, see “STRING table” on page 360

Table 13. Table summary (continued)

Category	Table name	Description
	SI	a string interface, the interfaces on the top of a string: A0, A1 ..., see “String interface (SI) table” on page 362
	UNIT	a physical unit of a string, see “UNIT table” on page 363
	DEVICE	an IODF I/O device, see “DEVICE table” on page 363
Crossbar switch related	**XBAR**	a physical crossbar switch, see “XBAR table” on page 364
	XBARCFG	a named crossbar configuration describing the crossbar connectivity, see “XBARCFG table” on page 365
	XCI	a crossbar channel interface (not read by the import function), see “Crossbar channel interface (XCI) table” on page 366
	XCUI	a crossbar control unit interface (not read by the import function), see “Crossbar control unit interface (XCUI) table” on page 366
Patchport related and general box port related	**CABINET**	a patchport cabinet or general box, see “CABINET table” on page 366
	PANEL	a patchport or general box port panel, see “PANEL table” on page 367
	PATCHPRT	a patchport or general box port, see “PATCHPRT table” on page 367
Connection related	**CABLE**	a physical cable, see “CABLE table” on page 369
	CONVERTR	a converter, see “CONVERTR table” on page 368
	LOGLINK	describes the end points of a logically complete series (see “LOGLINK table” on page 370). Useful for reporting purposes. See “LOGLINK table” on page 370.
	CONNECT	a connection between two objects, see “CONNECT table” on page 373
	PPRCLINK	a PPRC link/PPRC connection, see “PPRCLINK table” on page 376
Logical connection related	**DEVCU**	device - control unit connections, see “DEVCU table (export-only)” on page 377
	CUCHPID	control unit - CHPID connections, see “CUCHPID table (export-only)” on page 377

Table 13. Table summary (continued)

Category	Table name	Description
	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 378

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 14. 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 PC 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 PC configuration file (they have already been placed).

Table 14. PROC Table (continued)

Field Name	Format	Description
Style	Enum	Applicable only for new processors that are not already in the PC 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 itself): • 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 PC configuration file.
Status	Enum	Applicable only for new processors that are not already in the PC 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 itself): • 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 PC configuration file.

Table 14. PROC Table (continued)

Field Name	Format	Description
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). If you export, the same value will be shown 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.
ufi01	Text	User field 1, or customized user field name.
ufi02	Text	User field 2, or customized user field name.
...	...	...
ufi20	Text	User field 20, or customized user field name.
Local sysname	Text EO	Value taken from the IODF object. Blank if not applicable.

PART table

Table 15. 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 dot, followed by a number for the CSS ID (for XMP processors only), followed by the partition name. For reserved partitions, also specify the MIF ID in parenthesis, for example, PROC2084.2.* (5).
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.

Table 15. PART Table (continued)

Field Name	Format	Description
Usage	Text EO	Value taken from the IODF object. For example, CF.
ufi01	Text	User field 1, or customized user field name
ufi02	Text	User field 2, or customized user field name
...	...	...
ufi20	Text	User field 20, or customized user field name

CHPID table

Table 16. 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 dot (.), followed by a number for the CSS ID (for XMP processors only), followed by 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.
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, IOC, OSA.
PCHID	Text EO	Value taken from the IODF object. Applicable only for XMP processors. Blank if not applicable.
Spanned with	Text EO	Value taken from the IODF object. Only applicable for spanned CHPIDs. Lists all channel subsystems this CHPID is spanned with, separated by blanks. Example: `PR0C1.1  PR0C1.2`

Table 16. CHPID Table (continued)

Field Name	Format	Description
>160 TCP/IP stacks	Text EO	Value taken from the IODF object. Indicates if the CHPID supports queue prioritization. Either empty or Yes.
HCA ID	Text EO	ID of the host communication adapter. Value taken from the IODF object. Blank if not applicable.
HCA Port	Text EO	Port on the host communication adapter. Value taken from the IODF object. Blank if not applicable.
Maximum frame size	Text EO	Maximum frame size of an IQD channel. Value taken from the IODF object. Blank if not applicable.
IQD function	Text EO	IQD function of an IQD channel. Value taken from the IODF object. Blank if not applicable. Valid values are: Basic Hipersockets, IEDN Access (IQDX), External Bridge.

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 17. 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 PC configuration. The label which should be applied to the internal controller of this switch. Ignored for switches which are already in the PC configuration file.
String Label	Text	Applicable only for new switches that are not already in the PC configuration. The label which should be applied to the internal string of this switch. Ignored for switches which are already in the PC configuration file.
ufi01	Text	User field 1, or customized user field name
ufi02	Text	User field 2, or customized user field name
...	...	...
ufi20	Text	User field 20, or customized user field name

PORT table

Table 18. 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 19. 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 PC object), this field is 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 PC configuration. Ignored for controllers which are already in the PC 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 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 controllers that are not already in the PC configuration. Ignored for controllers which are already in the PC 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 in the PDF Name column.

Table 19. CONTRLLR Table (continued)

Field Name	Format	Description
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 PC configuration. Ignored for controllers which are already in the PC 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.
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

Table 19. CONTRLLR Table (continued)

Field Name	Format	Description
Grouped with	Text	Applicable only for new controllers that are not already in the PC 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 PC configuration file (they have already been placed).
ufi01	Text	User field 1, or customized user field name
ufi02	Text	User field 2, or customized user field name
...	...	...
ufi20	Text	User field 20, or customized user field name

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 one interface on a 3990. This is a known instance where an export, followed by an import into an empty file, will not re-create the original configuration exactly.

Controller channel interface (CCI) table

Table 20. CCI Table

Field Name	Format	Description
Table Name	**Text**	**CCI**

Table 20. CCI Table (continued)

Field Name	Format	Description
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 21. 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: `6100 #5/Bus 0/3   (interface label omitted)` `5500 #3/Y-Bus/6/1 (second unnamed interface` `on adapter 6)` 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 22. CU Table

Field Name	Format	Description
Table Name	**Text**	**CU**
ID	Hex	<IODF Key> The ID of the IODF object.

Table 22. CU Table (continued)

Field Name	Format	Description
Container	Text	Applicable only for new control units that are not already in the PC 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 PC 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.
SSID	Text	The subsystem ID of the control unit.

STRING table

Table 23. STRING Table

Field Name	Format	Description
Table Name	**Text**	**STRING**

Table 23. STRING Table (continued)

Field Name	Format	Description
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` `0400-0000` `0400-0000(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 PC configuration. Ignored for strings which are already in the PC 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 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 PC configuration. Ignored for strings which are already in the PC 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 in the PDF Name column.

Table 23. 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 PC configuration. Ignored for strings which are already in the PC 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.
ufi01	Text	User field 1, or customized user field name
ufi02	Text	User field 2, or customized user field name
...	...	...
ufi20	Text	User field 20, or customized user field name

String interface (SI) table

Table 24. 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 0001-0000/A0, 0400-0000/A0, or 0400-0000(MVS1)/A0.

Table 24. SI Table (continued)

Field Name	Format	Description
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 25. 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.
ufi01	Text	User field 1, or customized user field name
ufi02	Text	User field 2, or customized user field name
...	...	...
ufi20	Text	User field 20, or customized user field name

DEVICE table

Table 26. DEVICE Table

Field Name	Format	Description
Table Name	**Text**	**DEVICE**

Table 26. DEVICE Table (continued)

Field Name	Format	Description
ID	Text	<IODF Key> The ID of the IODF object. This is normally the device number, followed by a four-digit hexadecimal suffix, appended with a '-', which helps to distinguish devices with duplicate IDs. Also, to help ensure uniqueness, the device number is followed by the name, enclosed in parenthesis, 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: `0400-0000` `0400-0000(MVS1)`
Container	Text	Applicable only for new devices that are not already in the PC 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 is placed into that unit. If a string tag is specified, the device is placed into the first available position in that string using HCM's normal heuristics. If this field is not specified, HCM places 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 PC 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.
PPRC Usage	Text EO	Value taken from the IODF object. The PPRC Usage type of the device.

XBAR table

Table 27. 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.
ufi01	Text	User field 1, or customized user field name
ufi02	Text	User field 2, or customized user field name
...	...	...
ufi20	Text	User field 20, or customized user field name

XBARCFG table

Table 28. 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 29. 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 30. 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 31. CABINET Table (also containing general boxes)

Field Name	Format	Description
Table Name	**Text**	**CABINET**
Id	Text	This serves as the identifier for cabinets or general boxes, since HCM enforces that it is unique.
Description	Text	The description of the cabinet or general box.
Model	Text	Applies to cabinets only. 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 (applies to cabinets only).
Graphical Position	Enum	One of HCM's values for "Graphical Position". Valid values are: near processors, central (this is the default), near devices.
ufi01	Text	User field 1, or customized user field name
ufi02	Text	User field 2, or customized user field name
...	...	...

Table 31. CABINET Table (also containing general boxes) (continued)

Field Name	Format	Description
ufi20	Text	User field 20, or customized user field name
Container Type	Text	Contains 'Yes' if the object is a general box, contains 'No' if the object is a cabinet.
Long Description	Text	Contains the text from the 'Comments' field in the 'Create/Edit General Box' dialogs (applies to general boxes only).

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 32. PANEL Table

Field Name	Format	Description
Table Name	**Text**	**PANEL**
ID	Text	This consists of the cabinet ID or general box name followed by a slash "/" and a panel ID. If a panel of this ID does not exist in the cabinet or general box, it will be added to the bottom of the cabinet or general box.
Description	Text	The description of the panel.
Naming format	Text	The format string used to generate patchport or general box port 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 or general box ports 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 or general box ports in the panel. Used only if the panel is being created; ignored if the panel already exists. This field must be specified.
Sequenced row first	Bool	If Yes, patchports or general box ports are sequenced in a row-first manner when displayed in lists or used in operations involving multiple patchports or general box ports.

PATCHPRT table

Table 33. PATCHPRT Table

Field Name	Format	Description
Table Name	**Text**	**PATCHPRT**

Table 33. PATCHPRT Table (continued)

Field Name	Format	Description
Name	Text	The name of the patchport or general box port as generated by the naming format for this cabinet or general box. Resolution of patchport or general box port names is made after all cabinets or general boxes 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" or "Table Name:General Box Port Name".
Cabinet	Text EO	The ID of the cabinet which contains this patchport, or the name of the general box which contains this general box port.
Explicit Label	Text	A name specified by you that is explicitly assigned to the patchport or general box port (overriding the name which would be assigned with the naming format).
Installed	Bool	Yes, if the group containing this patchport or general box port should be marked as installed. There is no default for this field. When no data are present here, the installed status is left unchanged. When panel spaces are created, all groups are marked as installed. Since patchports/general box ports are installed in groups, export will place the same value in this column for all patchports/general box ports in the group. On import, all patchports/general box ports 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 with 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.
Position	Enum	One of HCM's allowed values for position. Valid values are: `Near processor`, `Above switch`, `Below switch`, `Near device`.

CONVERTR table

Table 34. 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.

Table 34. CONVERTR Table (continued)

Field Name	Format	Description
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.
ufi01	Text	User field 1, or customized user field name
ufi02	Text	User field 2, or customized user field name
...	...	...
ufi20	Text	User field 20, or customized user field name

CABLE table

Table 35. CABLE Table

Field Name	Format	Description
Table Name	**Text**	**CABLE**
ID	Text	The cable identifier.

Table 35. CABLE Table (continued)

Field Name	Format	Description
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** Bidirectional **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 36. LOGLINK Table

Field Name	Format	Description
Table Name	**Text**	**LOGLINK**

Table 36. LOGLINK Table (continued)

Field Name	Format	Description
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 37. 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 375
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.

Table 37. CONNECT Table (continued)

Field Name	Format	Description
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.
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.

Table 37. CONNECT Table (continued)

Field Name	Format	Description

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.

For connections on the same level (for example, between patchports), there is no defined order. The source/target association is defined according to the interface that you selected in order to establish the connection.

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.

Note:

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 370 under "Connections from CHPIDs on OPPP processors" on page 373. 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 PC 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 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 38. 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.

Table 38. PPRCLINK Table (continued)

Field Name	Format	Description
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 39. 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 40. 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.
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 41. 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.
Subchannel Set ID	Text EO	The ID of the subchannel set where the device is placed related to the device-processor definition.

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 stand-alone 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 stand-alone mode, you will see an appropriate error message indicating that the attempted operation is not permitted in stand-alone mode. The name of the table currently being exported appears in the status bar, so you will know which table required host access.

Chapter 15. Physical description files (PDF)

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* (PDF) 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 PC 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 56). 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.

Note:

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 quotation marks.

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.

© Copyright IBM Corp. 1995, 2021

An ellipsis (...) indicates that multiple entries of the type immediately preceding the ellipsis are allowed.

Parentheses (not-bold) 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 42. 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. The default value is 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 43. 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.

Table 43. String PDF Attributes (continued)		
Attribute	**Allowed Values**	**Description**
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 16. 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 about how to proceed in handling a problem can be found in

- *z/OS MVS Diagnosis: Reference*
- *z/OS MVS Diagnosis: Tools and Service Aids*
- *z/OS Problem Management*
- *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 386
- "Controlled exit at the workstation" on page 386
- "Communication problems using TCP/IP under z/OS" on page 386
- "Communication problems using TCP/IP under z/VM" on page 387
- "Message problems" on page 387
- "Incorrect results" on page 387
- "Incorrect configuration diagram" on page 388
- "Problems printing reports/configuration diagrams" on page 388
- "Abnormal termination of HCD host" on page 388
- "HCD host problems" on page 389
- "Problems when using the RMF DDS" on page 389

© Copyright IBM Corp. 1995, 2021

- “Problems with HCM User's Guide” on page 390
- “Exporting an MCF” on page 390
- “Incorrect help information” on page 390

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 about identifiers, refer 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.

Controlled exit at the workstation

If HCM detects a "logic error", it terminates 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 PC file (see “The HCMERR.LOG PC file” on page 392)

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 log in with HCM using a wrong user ID or password: You get an EEQX401E 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.

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 Chapter 17, “HCM messages,” on page 395.

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 10)?
- 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.

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

- PC 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 391 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:

- 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 8),
- 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 433).

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 the RMF DDS

When trying to access performance data for HCM objects, you may run into problems with using the RMF DDS. The following messages may appear:

- **HCM is not able to determine the RMF Distributed Data Server (DDS).**

Table 44. DDS problem handling (1)

Problem reason	User response
The RMF DDS is not yet started.	Start the RMF DDS.

Table 44. DDS problem handling (1) (continued)	
Problem reason	**User response**
A wrong version of the RMF DDS is running or the DDS is running with an incorrect configuration.	Ensure that the RMF DDS is the one for z/OS 1.8 RMF.
A wrong version of HCD is running.	To use the new functions of z/OS 1.8 HCM, you must run z/OS 1.7 HCD with APAR OA14334.

- **The connection to the RMF Distributed Data Server (DDS) failed.**

Table 45. DDS problem handling (2)	
Problem reason	**User response**
The TCP/IP connection between HCM and DDS failed.	Check your network authorization and authentication mechanisms, because these might block the communication. For example, the HCM workstation may not be allowed to access the required DDS port. This may be caused by a personal firewall or a network firewall.

For more information about the RMF Distributed Data Server, refer to in the .

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)

Exporting an MCF

Action **Utilities ---> Export Master Configuration File** is only provided for reporting MCF problems to IBM and is not required for normal use of MCF sharing. IBM Support may ask you to create an exported MCF and send it in for problem diagnosis. For this purpose, this action lets you export the current configuration's associated MCF, as it is stored on the host, to a file on the PC.

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

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 about the z/VM HCD TCP/IP dispatcher for HCM trace see the following section “z/VM HCD TCP/IP Dispatcher for HCM commands” on page 391.

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 PC file

HCM incorporates another aid for problem determination. If a severe or unexpected error or a controlled exit occurs, HCM saves its current processing 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 severe error or a controlled exit occurs. You can truncate or delete the file. If you do not delete the file, the data of the processing environment is appended for each such severe error or controlled exit.

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, nnnn, s) 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/cccnnnns	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 about these steps, refer to *z/OS MVS Diagnosis: Reference* 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 PC 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 configuration files.

  In certain cases IBM might ask you for the configuration file and associated IODF in order to recreate the problem.
- An HCD trace file of the problem. See “Using the HCD trace facility” on page 391 for more information.
- The HCMERR.LOG file, if it exists. Refer to “The HCMERR.LOG PC file” on page 392 for details.

Be prepared to recreate the error and, if possible, to give IBM copies of your IODF and configuration file (.HCM).

Chapter 17. 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 **Controlled Exit. 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 393 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.

© Copyright IBM Corp. 1995, 2021

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 can be created and written.

EEQ0209E **Error trying to save PWS config file *filename*. The original 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 can 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 can be created and written.

EEQ0212E **Operation not allowed in Stand-alone Mode.**

Explanation

Connecting and disconnecting interfaces, as well as operations which require communication with the host, are not allowed when running HCM in stand-alone Mode.

System action

Aborts the attempted operation.

User response

Open the PWS file again when HCM is not in Stand-alone 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.

EEQ0216E **Operation is not allowed for a MUA-enabled IODF.**

Explanation

The accessed IODF is enabled for multi-user access (MUA). You cannot apply updates to MUA-enabled IODFs using HCM.

System action

Aborts the attempted operation.

User response

You can apply physical-only definitions to the configuration. In addition, you can work with the IODF in read-only mode, for example, view the configuration, or generate reports. You can use an HCD version providing MUA support to update the IODF. In order to allow HCM to update the IODF, multi-user access has to be disabled for the IODF in HCD.

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 380.

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 382.

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 16, “Problem determination,” on page 385 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 16, “Problem determination,” on page 385 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 16, “Problem determination,” on page 385 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 readme.rtf 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 readme.rtf 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 Stand-alone 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 Stand-alone 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 stand-alone 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 stand-alone 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 393 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 393 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 393 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 393 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 393 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 393 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 393 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 is correctly installed.**

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 host.

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 393 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 393 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 393.

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.

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.

EEQX104E **Host communication error.**

Explanation

The HCM dispatcher/agent on host <name> might not be AT-TLS configured for this session on port <port#>

System action

No connection to host.

User response

Check whether the HCM dispatcher/agent is AT-TLS configured on the host, or check whether the root CA certificate is installed in Trusted Root Certification Authorities on the workstation. See “Encrypting the

host communication for z/OS" on page 19 for details on how to configure the host and the workstation.

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 your HCM user id and your password.

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 A. 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 processors that are 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 **Create...** button. The **Create Processor** dialog allows you to create an alternate processor under the selected machine. Note that you can only create an SMP shadow in an SMP machine or an XMP shadow in an XMP machine. Corresponding channel subsystems are created as well. Combinations of XMP processorsand SMP processors within a machine are not allowed.

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 displays 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.

Both primary and alternate XMP processors, must have the same number of channel subsystems with the same CSS IDs. If you delete or add a channel subsystem to or from a primary processor, the same updates are performed on its alternate shadow processors, too.

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/

© Copyright IBM Corp. 1995, 2021

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 automatically appears 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 attempts to assign each processor to a machine by matching first serial numbers, then SNA addresses. If unsuccessful, HCM tentatively assigns the new processor to a new machine. When the **Machine** dialog automaticaly appears, it asks 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 415.

At this point, HCM identifies new processors and labels them as `new`. HCM also identifies deleted processors and labels them as `deleted`.

Appendix B. 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.

© Copyright IBM Corp. 1995, 2021

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.

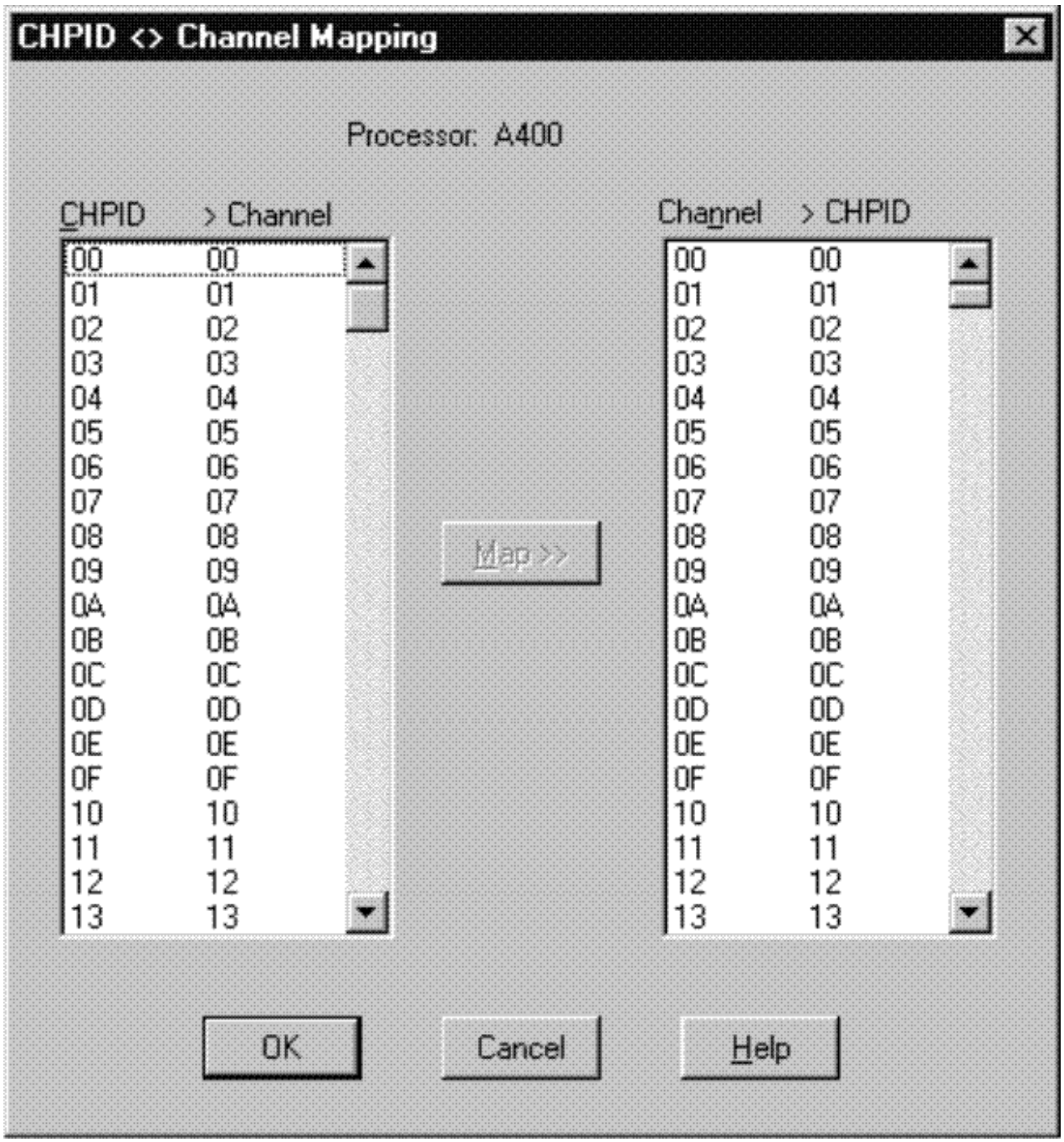


Figure 196. CHPID <> Channel Mapping Dialog

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 C. 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.

© Copyright IBM Corp. 1995, 2021

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

Object	Formatting convention (NAME plus additional information)	Examples
Converter	ConverterName 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 is presented 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 D. Menu bar choices summary

The menu bar choices are acting either on the *configuration work (*.hcm/hcmz)* file or on the *configuration production (*.hcr/hcrz)* file.

For all choices, the following tables indicate in column **Std-alone** whether the function is also available in stand-alone mode. However, even if a function is indicated as being available, be aware that you can apply the function only to physical aspects of the configuration, and not to logical aspects that would impact the associated IODF, which is not accessed in stand-alone mode. Hence, all function descriptions in the following tables that relate to an IODF do not apply for stand-alone mode.

File menu

Choice	Function	Std-alone
Open...	Open an existing configuration (.HCM, .HCR) file and the corresponding IODF.	y
Open Specia**l**...	Open an existing configuration (.HCM, .HCR) file and a specified IODF.	n
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.	y
Close	Save the configuration (.HCM, .HCR) file and the corresponding IODF, and then close them.	y
IODF **D**etails...	Display important IODF information.	y
Copy Co**n**figuration Files...	Copy the configuration (.HCM, .HCR) file to a new file and the corresponding IODF to a new data set.	n
Build **P**roduction...	Create a production IODF.	n
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 PC.	n
Set **A**ccess Mode	A dialog lets you select one of the following access modes for the current configuration: • **Single-user mode - Local configuration** disables MCF sharing. This operation deletes the existing MCF from the host. Synchronization of physical updates on multiple local copies is now in the user's responsibility. • **Master configuration file mode** enables the configuration for MCF sharing. This choice creates a master configuration file on the host which allows several HCM users to work cooperatively and safely on a single configuration, sequentially sharing the configuration in connected mode.	n
IODFs...	Open a dialog to manage IODFs.	n
View IODF Reports...	View HCD I/O configuration reports and save the output in a workstation file.	n

© Copyright IBM Corp. 1995, 2021

Choice	Function	Std-alone
View IODF Compare Reports...	View HCD IODF Compare Reports and save the output in a workstation file.	n
View IODF Compare CSS/OS Report...	View HCD IODF CSS/OS Compare Reports (comparing devices within different CSSs/OSs in one IODF) and save the output in a workstation file.	n
Compare **H**CM Files...	Compare two HCM configuration files with respect to information contained in the associated IODF (logical view) and additionally with respect to physical information like cable connections, patchports, controllers, and so on. The output is presented in a window and you can save it in a file.	y
Page Se**t**up...	Setup the page for printing purposes; setup the printer and its parameters.	y
Print Dia**g**ram...	Print the configuration diagram.	y
Generate HCM **R**eports...	Set parameters for the reports that you are going to create and/or print. Also, you can specify a report title, cover page notes, and a table of contents. You can determine the report contents and select the objects for reporting, and you can print an index.	y
Print Cable L**a**bels...	Print cable labels to a text file or printer.	y
I**m**port Data...	Import HCM data into a configuration.	n
Export Data...	Export configuration data to a text file in a directory.	y
E**x**it (Alt+F4)	Save and close the configuration (.HCM, .HCR) file and the corresponding IODF and exit HCM.	y

Edit menu

Choice	Function	Std-alone
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".	y
Delete/Disconnect selected object (Del)	Delete the most recently selected object(s) or disconnect the most recently selected interface(s) 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.	y
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.	y
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.	y
Processor...	Edit a particular processor.	y

Choice	Function	Std-alone
P**a**rtition...	Edit a particular partition.	y
C**H**PID...	Edit a particular CHPID.	n
Sw**i**tch...	Edit a particular switch.	y
P**o**rt...	Edit a particular port.	n
Crossbar S**w**itch...	Edit a particular crossbar switch.	y
Controller...	Edit a particular controller.	y
Control **U**nit...	Edit the controller containing a particular control unit.	y
String...	Edit a particular string.	y
De**v**ice...	Edit a particular device.	n
Con**v**erter...	Edit a particular converter.	y
Patchpo**r**t...	Edit the cabinet containing a particular patchport.	y
Cabi**n**et...	Edit a particular cabinet.	y
Genera**l** Box Port...	Edit a particular port of a general box.	y
General Bo**x**	Edit a general box.	y
Ca**b**le Assignments...	Edit the cable assignments to connections.	y
OS Confi**g**urations...	Edit the OS configurations.	n
C**T**C connections...	Edit CTC connections.	y
Configuration Pac**k**ages...	Open the ***Configuration Packages*** dialog for an accessed work IODF to create and edit configuration packages. Open this dialog for an accessed production IODF to transmit configuration packages defined from this IODF.	n
C**F** connections...	Edit CF connections.	y
PPRC conn**e**ctions...	Edit Peer-to-Peer Remote Copy connections (PPRC connections).	y
Order Processors...	Change the order of processors in the configuration diagram.	y
Titles & **F**ootnotes...	Edit the title and footnotes that appear in configuration diagrams and reports.	y
Manage Shadow Processors...	Edit the assignment of alternate (shadow) processors to physical machines.	y

View menu

Choice	Function	Std-alone
Previous View (F12)	Display the previous automatically saved view of a configuration diagram.	y

Choice	Function	Std-alone
Fit to Window (Ctrl+Home or F2)	Reduce the scale of the configuration diagram to fill the window.	y
Zoom Out (Home)	Reduce the scale of the configuration diagram by one step.	y
Zoom In (End)	Enlarge the scale of the configuration diagram by one step.	y
Maximum Size (Ctrl+End)	Enlarge the scale of the configuration diagram to the maximum size	y
Filter Diagram...	Specify which objects and connections are included in the diagram.	y
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.	y
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.	y
Visualized Matrices	Define a configuration for each ESCON or FICON switch or crossbar switch to be visualized.	y
Show selected connections	Show connections from multiple selected interfaces.	y
Hide selected connections	Hide connections from multiple selected interfaces.	y
Restore Named View (F3)	Select a saved named view to be shown as active configuration diagram.	y
Save Named View	Save the displayed view.	y
Delete Named View	Delete a named view.	y
Restore Matrix Set	Select a saved named matrix view to be shown in the configuration diagram.	y
Save Matrix Set	Save the current matrix set.	y
Delete Matrix Set	Delete a named matrix set.	y
Text Detail...	Specify which text appears in the configuration diagram.	y
Preferen**c**es...	Set certain preferences for the configuration diagram.	y
Colors...	Enable/disable and set colors for visual warnings for logical/physical inconsistencies.	y
Ob**j**ect Information Bar	Include or exclude the Object Information Bar.	y
Stat**u**s Bar	Include or exclude the Status Bar.	y
Show Page **B**reaks	Display the effects of page breaks and printer margins.	y

Locate menu

Choice	Function	Std-alone
Se**l**ected object (F8)	Locate the most recently selected object. This menu item's text changes to describe the most recently selected object.	y

Choice	Function	Std-alone
System	Locate a system in a sysplex. This menu item's text changes to describe the most recently selected object.	n
Processor...	Locate a particular processor	y
Partition...	Locate a particular partition	y
CHPID...	Locate a particular CHPID	y
Switch...	Locate a particular switch	y
Port...	Locate a particular port.	y
Crossbar Switch...	Locate a particular crossbar switch	y
Controller...	Locate a particular controller.	y
Control Unit...	Locate the controller containing a particular control unit.	y
String...	Locate a particular string.	y
Device...	Locate a particular device.	y
Converter...	Locate a particular converter.	y
Patchport...	Locate the cabinet containing a particular patchport.	y
Cabinet...	Locate a particular cabinet.	y
General Box Port...	Locate a particular port of a general box.	y
General Box...	Locate a particular general box.	y
CTC connection...	Locate a particular CTC connection.	y
CF connection...	Locate a particular CF connection.	y
PPRC connection...	Locate a particular Peer-to-Peer Remote Copy connection (PPRC connection).	y
Cable...	Locate a particular cable.	y

Create menu

Choice	Function	Std-alone
Processor...	Create a new processor.	n
Switch...	Create a new switch.	n
Crossbar Switch...	Create a new crossbar switch.	y
I/O Subsystem...	Create a new or copy an existing I/O subsystem.	n
Controller...	Create a new controller.	n
String...	Create a new string.	n

Choice	Function	Std-alone
Cabi**n**et...	Create a new cabinet.	y
General Bo**x**...	Create a new general box.	y
C**T**C connection...	Create a new CTC connection.	n
C**F** connection...	Create a new CF connection.	n
PPRC conn**e**ction...	Create a Peer-to-Peer Remote Copy connection (PPRC connection).	y

Utilities

Choice	Function	Std-alone
A**u**toconfiguration...	Automatically discover and configure storage devices.	n
Move Port Connections...	Move port connections from a switch to another or within a single switch.	n
Aggregate CHPIDs...	Aggregate several CHPID definitions onto one single CHPID.	n
Convert ESCON Port to **F**C Port...	Migrate CNC channels to fiber channels.	n
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.	n
CHPID Mapping **T**ool Support...	Invokes a dialog to interact with the CHPID Mapping Tool to assign missing PCHIDs to CHPIDs (or update existing PCHID-CHPID assignments) for XMP processors. See "Using the CHPID Mapping Tool Support utility" on page 73 for further information.	n
Export **I**OCP or I/O Configuration statements	Invokes a dialog that lets you • export an IOCP input file for a selected processor • or export a certain type of configuration statements (for processor, operating system or switch) either into a z/OS data set on the host or into a file on the workstation.	n
Check Configuration File...	Invokes a dialog to check configuration files for structural consistency. See "Checking a configuration" on page 76 for further information.	y
Export Master Configuration File	Exporting an MCF-enabled configuration. This operation exports the MCF, as it is stored on the host, to a file on the PC. If required, you can send this file to IBM Support to diagnose MCF-related problems.	n
Browse Performance Data...	Invokes a utility that lets you examine performance data of objects in your active sysplex, for example, to discover potential bottlenecks.	y

Choice	Function	Std-alone
WWPN Prediction Tool Support...	Invokes a dialog that lets you • Export FCP data from a selected processor into a specified output file • Launch the WWPN Prediction Tool (using the output file with the exported FCP data as input) to assign world-wide port names to virtualized FCP ports for configuring SAN devices.	n

Options menu

Choice	Function	Std-alone
Confirmation...	Instruct HCM whether to prompt for a confirmation before deleting an object or disconnecting an interface.	y
File Comp**r**ession...	Choose whether to store the HCM configuration files in a compressed or uncompressed format. Compressed configuration files use less disk space. They cannot be opened with HCM versions prior to z/OS V1R9.	y
Field Names...	Assign customized names to the user data fields.	y
Interface Selection...	Choose whether to select the object containing an interface or its cable when an interface is selected with the mouse.	y
Activate HCD **T**race	Turn the HCD trace facility on or off.	n
No **D**iagram Mode	Invoke a mode where the diagram is not shown.	y
Retain **U**nused Connections	You can turn on or off Retain Unused Connections during HCM session by toggling the Retain Unused Connections command in the Options menu. HCM disconnects the connections between switch port and control unit by default, when removing logical CHPID definitions from the control unit. If you want to retain the connections between the control unit and the switches, you can enable this command in advance.	n

Help menu

Choice	Function	Std-alone
Index	Display the main HCM help topic index.	y
Keyboard	Describe the use of the Function keys and various cursor control key shortcuts.	y
About...	Open a window describing the current release of HCM.	y

Appendix E. 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 EEQHCM.INI file to access optional parameters for the login request (see also “Purpose and location of the HCM INI file” on page 5).

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 quotation marks, 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.

© Copyright IBM Corp. 1995, 2021

[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

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 system log (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 Chapter 17, “HCM messages,” on page 395.

Appendix F. Accessibility

Accessible publications for this product are offered through IBM Documentation (www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the Contact the z/OS team web page (www.ibm.com/systems/campaignmail/z/zos/contact_z) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features in z/OS can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Assistive technology products such as screen readers function with the user interfaces found in z/OS. Consult the product information for the specific assistive technology product that is used to access z/OS interfaces.

Keyboard navigation of the user interface

You can access z/OS user interfaces with TSO/E or ISPF. The following information describes how to use TSO/E and ISPF, including the use of keyboard shortcuts and function keys (PF keys). Each guide includes the default settings for the PF keys.

- *z/OS TSO/E Primer*
- *z/OS TSO/E User's Guide*
- *z/OS ISPF User's Guide Vol I*

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM Documentation with a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line because they are considered a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that the screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1)

© Copyright IBM Corp. 1995, 2021

are mutually exclusive alternatives. If you hear the lines `3.1 USERID` and `3.1 SYSTEMID`, your syntax can include either `USERID` or `SYSTEMID`, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The `*` symbol is placed next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element `*FILE` with dotted decimal number 3 is given the format `3 \* FILE`. Format `3* FILE` indicates that syntax element `FILE` repeats. Format `3* \* FILE` indicates that syntax element `* FILE` repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol to provide information about the syntax elements. For example, the lines `5.1*, 5.1 LASTRUN, and 5.1 DELETE` mean that if you use more than one of the `LASTRUN` and `DELETE` syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, it indicates a reference that is defined elsewhere. The string that follows the % symbol is the name of a syntax fragment rather than a literal. For example, the line `2.1 %OP1` means that you must refer to separate syntax fragment `OP1`.

The following symbols are used next to the dotted decimal numbers.

? indicates an optional syntax element
The question mark (?) symbol indicates an optional syntax element. A dotted decimal number followed by the question mark symbol (?) indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example `5? NOTIFY`). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines `5 ?, 5 NOTIFY`, and `5 UPDATE`, you know that the syntax elements `NOTIFY` and `UPDATE` are optional. That is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

! indicates a default syntax element
The exclamation mark (!) symbol indicates a default syntax element. A dotted decimal number followed by the `!` symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the dotted decimal number can specify the ! symbol. For example, if you hear the lines `2? FILE, 2.1! (KEEP), and 2.1 (DELETE)`, you know that `(KEEP)` is the default option for the `FILE` keyword. In the example, if you include the `FILE` keyword, but do not specify an option, the default option KEEP is applied. A default option also applies to the next higher dotted decimal number. In this example, if the `FILE` keyword is omitted, the default `FILE(KEEP)` is used. However, if you hear the lines `2? FILE, 2.1, 2.1.1! (KEEP)`, and `2.1.1 (DELETE)`, the default option KEEP applies only to the next higher dotted decimal number, `2.1` (which does not have an associated keyword), and does not apply to `2? FILE`. Nothing is used if the keyword `FILE` is omitted.

*** indicates an optional syntax element that is repeatable**
The asterisk or glyph (*) symbol indicates a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line `5.1* data area`, you know that you can include one data area, more than one data area, or no data area. If you hear the lines `3* , 3 HOST, 3 STATE`, you know that you can include `HOST, STATE`, both together, or nothing.

Notes:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write `HOST STATE`, but you cannot write `HOST HOST`.
3. The `*` symbol is equivalent to a loopback line in a railroad syntax diagram.

+ indicates a syntax element that must be included

The plus (+) symbol indicates a syntax element that must be included at least once. A dotted decimal number followed by the + symbol indicates that the syntax element must be included one or more times. That is, it must be included at least once and can be repeated. For example, if you hear the line `6.1+ data area`, you must include at least one data area. If you hear the lines `2+, 2 HOST,` and `2 STATE`, you know that you must include `HOST, STATE,` or both. Similar to the `*` symbol, the + symbol can repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the `*` symbol, is equivalent to a loopback line in a railroad syntax diagram.

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Site Counsel
2455 South Road

© Copyright IBM Corp. 1995, 2021

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com®/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

© Copyright IBM Corp. 1995, 2021

Accessing licensed books on the web (z/OS)

z/OS licensed documentation in PDF format is available on the Internet at the Resource Link home page (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 Resource Link home page (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 receive confirmation that your request is being processed. After your request is processed you 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.

© Copyright IBM Corp. 1995, 2021

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 z/VM Internet Library (www.vm.ibm.com/library).

The IBM z/VM Internet Library also includes other information about z/VM, such as:

- Program directories
- Data areas and control blocks
- Monitor records

© Copyright IBM Corp. 1995, 2021

Glossary

This glossary defines technical terms and abbreviations used in the Hardware Configuration Manager documentation.

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.

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.

cabinet
Cabinets, also known as fiber management cabinets contain *panels* organized into *port groups* of *patchports*, which are pairs of fiber 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.

CIB
Coupling over InfiniBand. A channel path type to exploit InfiniBand technology for coupling connections.

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

© Copyright IBM Corp. 1995, 2021

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*.

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 file/config file
The *configuration file* (*.HCM file) contains HCM's interactive configuration diagram. You can modify and save different versions of your hardware configuration in a number of configuration files for historical and planning purposes. The configuration file is also called *work configuration file*.

The term *configuration files* refers to work configuration files (*.hcm/*.hcmz file) as well as to production configuration files (*.hcr/*.hcrz file). They are equivalent to the work and production IODFs, but are located on the PC. File extensions *.hcmz and *.hcrz indicate that the files are compressed.

See also *work configuration 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*.

CS5
A new channel path type.

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.

DDS
Distributed Data Server. A single data server on one system in the sysplex, which gathers the data from the RMF Monitor III distributed on all systems in the sysplex.

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.

D/R site OS configuration
HCD can automatically generate a disaster recovery (D/R) site OS configuration as a copy of the primary OS configuration. You need a D/R site OS configuration in a GDPS managed environment, where storage devices are mirrored over peer-to-peer remote copy (PPRC) connections in order to have a backup system defined for an emergency situation.

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.

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.

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.

general box
General box objects represent devices or connections that are not covered by other HCM objects so that you can have a complete view of the physical objects in your configuration. Examples for general boxes may be network devices and their connections or non-zSeries boxes. These general box objects appear as rectangle boxes on the right hand side in the configuration diagram. General boxes are not part of the logical definitions of an I/O configuration and thus are only stored in the HCM configuration file, but not in the associated IODF. Connections to and from a general box are defined via panels that contain general box ports.

See also *panel* and *general box port*.

general box port
A general box port designates a physical interface of a general box. With general box ports, you can connect general boxes into your configuration where required. The number and vertical and horizontal order of general box ports are defined within general box panels.

See also *general box* and *panel*.

HCA
With Coupling over InfiniBand, a peer coupling link is emulated on a Host Communication Adapter (HCA). See also *CIB*.

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.

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, 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.

jumper cable
Fiber *jumper cables* are usually used to make mutable connections between patchports.

LCSS
Logical channel subsystems. See also *channel subsystem*.

load [IODF load]
You can *load* IODF data from the host to the PC, for HCM to use when creating the interactive configuration diagram for that hardware configuration. The selected IODF file is loaded into a new HCM configuration on the workstation. If the selected IODF file has an associated MCF data set, then HCM creates the new configuration as a copy of the downloaded MCF. In this case, the new configuration can already contain physical definitions.

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*.

master configuration file (MCF)
The HCM master configuration file (MCF) is an HCM configuration stored on the host. It provides a central shared repository, allowing several HCD/HCM users to work on a single configuration cooperatively and safely.

MCF
See *master configuration file*.

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.

navigation dialog
Type of dialog allowing you to navigate to related dialogs.

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.

panel
Each cabinet or general box 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 fiber 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*.

PCIe
See *Peripheral Component Interconnect Express*.

peer-to-peer remote copy
Peer-to-peer remote copy (PPRC) connections are direct connections between DASD controller subsystems that provide a synchronous copy of a volume or disk for disaster recovery, device migration, and workload migration. 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.

Peripheral Component Interconnect Express (PCIe)
Native attached PCIe adapters provide software with a connection to many new functions. The hardware implementation consists of a fanout card that plugs into the CEC book. It uses a PCIe interface to connect to a switch card that is plugged into an I/O drawer. Each switch card controls a domain of eight I/O or PCIe cards.

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 PC. 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*.

reconciliation of VOLSERs
The term *reconciliation* describes the process of synchronizing the VOLSER values in the PC 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 configuration file was saved, the IODF file and configuration 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.

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.

Server Time Protocol link
A coupling facility connection which will be used as a timing-only link, providing the Server Time Protocol (STP) function. The STP is a time synchronization feature, which is designed to provide the capability for multiple System z9 and zSeries servers to maintain time synchronization with each other. STP is designed to allow events occurring in different System z9 and zSeries servers to be properly sequenced in time.

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.

stand-alone mode
This mode of operation of HCM allows you to open a configuration without a connection to the host.

STP link
See Server Time Protocol link.

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.

subchannel set
With a subchannel set you can define the placement of devices either relative to a channel subsystem or to an operating system. Depending on the processor type and the z/OS release, users can exploit additional subchannel sets in a channel subsystem.

This function relieves the constraint for the number of devices that can be accessed by an LPAR, because device numbers may be reused across all channel subsystems and subchannel sets. Depending on the machine implementation, the exploitation of the alternate subchannel sets is limited to certain device types.

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.

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.

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.

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.

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.

virtual channel identifier (VCHID)
For internal channel paths there does not exist a physical correspondence to hardware, therefore there does not exist a PCHID value. Instead, for internal reasons, a unique virtual channel ID (VCHID) is assigned during IML or after a dynamic "Add CHPID" request.

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 *configuration file*.

WWPN Prediction Tool
The worldwide port name (WWPN) prediction tool assists you with pre-planning of your Storage Area Network (SAN) environment. It assigns world-wide port names to virtualized FCP ports for configuring SAN devices. This stand alone tool is designed to allow you to set up your SAN in advance, so that you can be up and running much faster once the server is installed.

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

9034 converters 150
9035 converters 150

A

abnormal termination of HCD Host 388
About ... pull down choice 29
access list
    glossary definition 451
access rights for MCF data set 81
accessibility
    contact IBM 437
    features 437
activating configuration data 332
activity log
    naming convention 67
activity log, viewing 331
activity logging
    automatic 78
actual size
    glossary definition 451
adapter labels
    editing 136
adapters
    changing number 134
adapters, grouping 41
add a new controller 292
Add Device
    peer-to-peer remote copy 141, 147
    PPRC 141, 147
add four channels to an existing processor 285
add like function (HCD) 331
add some strings to the new controller 298
adding
    CHPIDs to a processor 122
    console to an OS configuration 253
    control units to controller segments 132
    devices to an esoteric 257
    devices to an esoteric within the OS configuration 251
    devices to an OS configuration 248
    patchport to fiber links 216
    physical data to the configuration 65
    user data to a cabinet 166
    user data to a controller 133
    user data to a crossbar switch 163
    user data to a partition 116
    user data to a processor 110
    user data to a string 143
    user data to a unit 146
adding and connecting new equipment 285
agent
    HCM agent job 433
    JCL for starting 411
aggregate CHPIDs 233
allowed switch port 160
alternate processor
    glossary definition 451
alternate processor definitions 415
alternate processor definitions, choosing 64
application error 386
arranging
    control units and connections in a controller 135
    controller components 134
assign a cable to the new connection 317
assign cables, other ways to 209
assign the control unit, choosing a controller segment to 139
assigning
    cables between the objects in a fiber link 216
    cables to connections 207
    CHPIDs to partitions 126
    control units to segments 139
    devices to units in the string 143
    partitions to CHPIDs 126
    PDFs to objects 56
    processors to machines during IODF load/resync 416
assistive technologies 437
attended mode
    applying configuration definitions 224
authorization file 15
AUTO_SS_DEVNUM_SCHEME
    map proposed control units 226
autoconfiguration
    attended mode 224
    discovered controllers 223
    unattended mode 224
autoconfiguration operation mode 224
autoconfiguration policy
    LP groups 229
    OS groups 230
automatic activity logging 78
automatic I/O configuration 221

B

backup files 6
backup IODF
    naming convention 66
backup your configuration 76
blocked ports, visualize 38
blocked switch port 160
browsing performance data 237
building
    configuration data sets 332
building a production IODF 72
building a work IODF 72
building CONFIGxx members 332
building IOCDSs 332
building your configuration 49
button
    glossary definition 451

C

cabinet
  glossary definition 451
cabinets
  adding user data to 166
  creating 166
  deleting 167
  editing 166
  modifying 165
cable descriptions 423
cable from a connection, deleting 210
cable labels
  how to print 276
  printed samples 276
  printing 211
cable types 210
cables
  assigning between the objects in a fiber link 216
  assigning to connections 207
  creating 208, 209
  deleting 209
  editing 208, 209
  in inventory 207
  in use 207
  managing 207
  types 210
  unused 207
cables to connections, assigning 207
candidate list
  glossary definition 451
candidate lists 115
CBDIODSP SECURITY command 17, 391
CBDIODSP TRACE command 391
CESTPATH commands 192
CF connections
  rules for moving 102
  rules for the Repeat scenario 106
CFReport 73, 451
change password 25
changing
  channel interface type 137
  number of string interfaces 145
  PDFs 379
  physical description (of strings) 144
  processor style 415
channel adapter
  glossary definition 451
channel interface
  connecting to another channel interface 192
  disconnecting 193
  glossary definition 451
channel interface from a CHPID, disconnecting 191
channel interface to a CHPID, connecting 184
channel interface type, changing 137
channel paths 117
channel subsystem
  copy 113
  create 113
  delete 114
  edit 113
  glossary definition 451
channel to channel 195
choosing a controller segment to assign the control unit 139
choosing a support level 109
choosing alternate processor definitions 64
CHPID
  glossary definition 452
  spanned 120
CHPID connections
  unused 202
CHPID Mapping Tool
  glossary definition 452
CHPID operation mode symbols 120
CHPID, managed/unmanaged 121
CHPIDs
  adding to a processor 122
  assigning to partitions 126
  deleting 126
  editing 126
  editing connected objects 122
  interface type 119
  mapping behavior 101
  modifying 117
  operation mode 120
  ordering objects connected to 122
  partition access 120
  selecting to view details 121
CHPIDs via coupling facility connections, connecting 203
CHPIDs via CTC connections, connecting 195
CHPIDs, channels, and CHPID mapping 418
CIB channel
  over-defining 125
CL5 channel
  defining 125
  editing 125
client
  HCM client 6
close command 78
closing your configuration 78
CMT 73
column definition record 334
combine CHPIDs 233
comma delimited ASCII file 211
commands
  CBDIODSP SECURITY 17, 391
  CBDIODSP TRACE 391
  for z/VM HCD TCP/IP dispatcher 17, 391
commands for z/VM HCD TCP/IP dispatcher for HCM 391
communication problem 386
communication requirements 4
compare
  CSS 279
  IODF 279
  IODFs 279
compare HCM configuration files 281
compress diagram horizontally 41
compressed configuration files 50
compressing configuration files 78
concurrent use (of IODF) by multiple users 52
config file
  glossary definition 452
configuration
  adding physical data to 65
  backup 76
  building 49
  closing 78
  connecting Objects 175

configuration *(continued)*
copying 76
creating a new one from scratch 66
master configuration file 79
modifying objects 99
opening an existing one 67
saving 78
configuration data
activating 332
processing 332
Configuration Data
defining/viewing with HCD 331
viewing/defining with HCD 331
configuration diagram
glossary definition 452
incorrect 388
printing 263
configuration file
deleting an existing one 78
glossary definition 452
configuration file and a new IODF, opening an existing one 68
configuration file and associated IODF, opening an existing one 67
configuration file and IODF, data integrity between 52
configuration file naming convention 78
configuration files
compare 281
compressing 78
resynchronization 53
configuration package
create 89
send to unattended node 93
transmit 92
configuration package objects 90
configuration packages 331
configuration port matrix 159
CONFIGxx members 332
connect a controller to the CHPID via the switch 318
connect a switch to CHPIDs 302
connect a switch to CHPIDs through the patchport cabinets 315
connect another switch to the other new CHPID 292
connect the controller to CHPIDs 293
connect the controller to the FICON switches 295
connect the new CHPIDs to patchports 288
connect the new switch to one of the new CHPIDs 291
connecting
"one processor per partition" processors 420
channel interface to a CHPID 184
CHPIDs via coupling facility connections 203
CHPIDs via CTC connections 195
controllers to controllers 191
controllers to processors 183
crossbar switches 218
devices to controllers 177
objects 175
objects in your configuration 175
string to a controller device interface 179
strings 179
switches to processors or other switches 193
connecting and adding new equipment 285
connection
CTC 197
connection targets
multiple selection 184, 195
connections
assigning cables 207
custom 180
hide and show 36
jumper 215
parallel 175
physical 179
serial 175
standard 179
connections on the diagram 35
connectivity report
CTC connection 199
connector lines 41
consoles
adding to an OS configuration 253
contact
z/OS 437
context diagram
glossary definition 452
control unit interfaces, ordering 165
control units
arranging in controllers 134
creating 139
deleting 139
editing 139
glossary definition 453
modifying 138
controlled exit at the workstation 386
controller adapter interface labels 136
controller adapter labels 136
controller channel interface
glossary definition 453
controller components, arranging 134
controller device adapter interface labels 136
controller device adapter labels 136
controller device interface, connecting the string to 179
controller interfaces
changing order 135
controller PDF syntax 380
controller segments, adding control units to 132
controller to processor
making a physical connection 185
making a standard connection 185
controllers
adding user data to 133
arrange control units and connections in 135
connecting to processors 183
creating 131
creating a string under 178
deleting 138
editing 132
glossary definition 452
grouping 137
modifying 130
ordering strings connected to 132
selecting an existing string 178
controllers, connecting to controllers 191
controlling text detail 41
converter
rules when moving connections 103
converter shape 41
converters

converters *(continued)*
9034 150
9035 150
creating 151
deleting 152
editing 152
modifying 150
replacing 153
converting to FICON channels 233
copy
channel subsystem 113
Repeat scenario 99
Upgrade scenario 99
copy wizard
mapping CHPIDs 100
copy wizards
concepts 99
copying
EDTs 255
esoterics 256
OS configurations 247
partitions 116
copying an existing IODF 71
copying an I/O subsystem 234
copying your configuration 76
coupling facility
glossary definition 453
coupling facility connections
creating 205
deleting 206
create a CTC connection 304, 306–308
create menu choices summary 429
creating
“one processor per partition” processors 417
cabinets 166
cables 208, 209
CHPIDs 118
configuration from scratch 66
control units 139
controllers 131
converters 151
coupling facility connection 205
crossbar switches 162
CTC connection 196
devices 147
EDTs 255
esoterics 256
I/O devices in a string 147
new configuration from IODF 56
new configuration from scratch 66
new processor under a machine 415
objects 45
OS configuration 246
partitions 115
processors 108
string under a controller 178
strings 141
switch configurations 158
switch's devices 147
switches 155
units 145
creating a PPRC connection 192
creating an I/O subsystem 234
cropped view 33
cropping the diagram 33
crossbar channel interface, disconnecting 219
crossbar channel interfaces, ordering 165
crossbar configurations, recording 163
crossbar control unit interfaces, editing controllers connected to 165
crossbar switch configuration
line styles 38
visualizing 38
crossbar switches
adding user data to 163
connecting 218
creating 162
deleting 165
disconnecting 219
editing 162
glossary definition 453
modifying 162
ordering controllers connected to 165
positioning on a daisy-chain 219
setting the size of 164
CSS
copy 113
create 113
delete 114
edit 113
CSS/OS
compare 279
CSS/OS compare reports 280
CTC connection
connectivity report 199
deleting 199
editing 197
CTC connection details
dialog 202
CTC connection report 202
CTC connections
creating 196
glossary definition 453
rules for moving 102
to “one processor per partition” machines 420
validating 201
CTC control units
unused 202
CUs 138

D

D/R OS configuration 246
D/R site OS configurations 246
daisy-chain, positioning a crossbar switch on a 219
daisy-chained connections, positioning objects 176
data
importing 337
data fields 29
data format 333
data format for export 333
data format for import 333
data space 389
dB loss 216
DDS
problem determination 389
dedicated connections, visualize 38
default matrix, generating 158

define CTC devices to OS configuration 308
define matrix set 38
define the devices to the operating system 300
define the fiber links for the connections 297
defining a CTC connection 302
defining an fiber cabling infrastructure 311
defining devices to OS configs 148
delete named matrix set 38
delete named view 37
deleting
  “one processor per partition” processors 417
  cabinets 167
  cable from a connection 210
  cables 209
  CHPIDs 126
  configuration file 78
  console from an OS configuration 254
  control units 139
  controller 138
  converters 152
  coupling facility connection 206
  crossbar switches 165
  CTC connection 199
  devices 249
  devices from an esoteric 258
  EDTs 255
  esoterics 256
  existing IODF 75
  I/O devices 149
  IODF 75
  multiple 46
  objects 46
  OS configurations 247
  partitions 117
  patchport from fiber links 216
  processor from a machine 415
  processors 110
  strings 145
  switch configurations 162
  switch's control units 140
  switches 158
  units 146
deleting a PPRC connection 193
deleting HCM 23
device <=> OS configuration parameters, editing 250
device adapter
  glossary definition 454
device candidate list 233
device candidate lists 115
device interface
  glossary definition 454
devices
  unused 202
devices to controllers, connecting 177
diagram
  compress horizontally 41
  connections 35
  drawing 33
  page breaks 40
  preferences 41
  printing 43
  saving 36
  scrolling 41
  tailoring 33
diagram *(continued)*
  zooming 41
diagram reports
  samples of 266
diagrams and reports 263
dialog
  glossary definition 454
dialogs 28
disconnecting
  channel interface from a CHPID 191
  crossbar channel interface 219
  permanent connections from patchports 214
  string interface from a device interface 183
  switch port from a target interface 195
disconnecting a channel interface 193
disconnecting an interface 46
discovered controllers
  autoconfiguration 223
discovery scope 222
dispatcher
  HCM dispatcher 411
dispatcher for HCM
  start and stop 26
dispatcher virtual machine 12
distributed data server
  problem determination 389
downloading HCM 4
drawing the diagram 33
dynamic activate 332
dynamic partition 115
dynamic switches 154

E

edit menu choices summary 426
editing
  “one processor per partition” processors 417
  cabinets 166
  cables 208, 209
  CHPIDs 126
  console for an OS configuration 254
  control units 139
  controllers 132
  controllers connected to crossbar control unit interfaces 165
  converters 152
  crossbar switches 162
  CTC connection 197
  device <=> OS configuration parameters 250
  EDTs 255
  esoterics 256
  fiber link containing a patchport 169
  fiber links 215
  Fiber links 212
  fiber links above the switch 157
  FICON links above the controller 137
  footnotes of text or diagram reports 265
  generics 261
  I/O devices 148
  infrastructure connections 212
  logical definitions between controller and processor 189
  objects 45
  objects connected to a CHPID 122
  OS configuration 247

editing *(continued)*
partitions 116
port matrix 159
processors 110
report title 265
strings 142
switch attributes 156
switch configurations 159
switch port attributes 157
switch ports in a configuration 162
switches 155
units 145
user-defined generics 261
editing a PPRC connection 192
editing an I/O subsystem 234
EDTs
copying 255
creating 255
deleting 255
editing 255
EEQ0000T 395
EEQ0200E 395
EEQ0201E 395
EEQ0202E 395
EEQ0203E 395
EEQ0204E 396
EEQ0205T 396
EEQ0206E 396
EEQ0207E 396
EEQ0208E 396
EEQ0209E 396
EEQ0210E 396
EEQ0211E 397
EEQ0212E 397
EEQ0213E 397
EEQ0214E 397
EEQ0215E 397
EEQ0216E 398
EEQ0250E 398
EEQ0251E 398
EEQ0252E 398
EEQ0253I 398
EEQ0260E 398
EEQ0261E 399
EEQ0262E 399
EEQ0263I 399
EEQ0400E 399
EEQ0401E 399
EEQ0402E 399
EEQ0403E 399
EEQ0404E 400
EEQ0405E 400
EEQ0410W 400
EEQ0411W 400
EEQ0412W 400
EEQ0413W 401
EEQ0414W 401
EEQ0415W 401
EEQ0416W 401
EEQ0600E 401
EEQ0601E 402
EEQ0602E 402
EEQ0603E 402
EEQ0604E 402
EEQ0605W 402
EEQ0606E 402
EEQ0607E 403
EEQ0608E 403
EEQ0609W 403
EEQ0610E 403
EEQ0611E 403
EEQ0991T 404
EEQ0992T 404
EEQ0993T 404
EEQ0994T 404
EEQ1000E 404
EEQ1001E 404
EEQ1002E 404
EEQ1003E 405
EEQ1100E 405
EEQ1101E 405
EEQ1102E 405
EEQ1103E 405
EEQ1104E 405
EEQ1105E 406
EEQ5000I 406
EEQ5001I 406
EEQ5003E 406
EEQ5004E 406
EEQ5005E 406
EEQ5006I 406
EEQ5007I 406
EEQ5008E 406
EEQ5010E 406
EEQ5011E 407
EEQ5012E 407
EEQ5013E 407
EEQ5014E 407
EEQ5015E 407
EEQ5016E 407
EEQ5017E 407
EEQ5018E 407
EEQ5019E 408
EEQ5021E 408
EEQ5022I 408
EEQ5023E 408
EEQ5024E 408
EEQ5025E 408
EEQ5026W 408
EEQ5027I 409
EEQ5028E 409
EEQ5029W 409
EEQ5030E 409
EEQ5031W 409
EEQ5032E 409
EEQ5050W 409
EEQ5051W 410
EEQ5052E 410
EEQ5053E 410
EEQC401E 410
EEQHCM.INI
defining ERRLOGDIR 392
EEQX100E 410
EEQX102E 410
EEQX104E 410
EEQX305E 411
EEQX306E 411
EEQX401E 411

EEQX405E 411
EEQX406E 411
EEQX407E 411
EEQX410E 411
EEQX414E 412
EEQX415E 412
EEQX416E 412
EEQX500E 412
EEQX501 412
EEQX502E 412
EEQX503E 412
eligible device tables 255
encrypt
  host communication 19
ERRLOGDIR 392
error log file 392
ESM 17
esoteric device groups 256
esoterics
  adding devices to 257
  copying 256
  creating 256
  deleting 256
  editing 256
  listing devices for 257
  removing devices from 258
existing cable for a connection, reusing 210
expired password 25
explicit device candidate list 183
export IODF 75
exported files
  column definition record 334
  header line 334
exporting
  IOCP or I/O configuration statements 235
exporting data
  table summary 347
exporting user data
  stand-alone mode 345
extended IQD 123
external security manager 17

F

FC port 233
feedback xxvii
fiber channels 233
fiber link
  disconnecting 216
  glossary definition 454
fiber link containing a patchport, editing 169
fiber links
  editing 215
Fiber links
  editing 212
fiber links above the controller, editing 137
Fiber links for point-to-point CTC connections 200
fiber management cabinets 165
FICON
  loopback connection 161
FICON bridge card 233
FICON bridge card, configuration changes 320
FICON devices
FICON devices *(continued)*
  I/O Autoconfiguration 221
file menu choices summary 425
filter diagram pull down choice 33
filter, locate objects 43
filter, view 33, 43
filtering objects 45
fit to window pull down choice 41
focusing a view 33
footnotes, edit 265
footnotes, for text and diagram reports 265
formatting conventions 421
full mode discovery 222
function keys 27

G

general box
  create 171
general box port naming format
  syntax of user-defined 170
  user-defined 170
general box ports 346
generate reports 265
generating
  diagram reports 264
  text reports 264
generic converters 150
generic device groups 258
generics
  editing a user-defined 261
  listing devices for system-defined 260
  listing devices for user-defined 261
  listing system-defined 259
  listing user-defined 260
GEOPARM control statements 192
granting I/O devices access to the new partition 115
graphical representation of physical objects 55
grouping adapters on controllers 41
grouping controllers into subsystems 137

H

handling problems 385
hardware requirements 3
HCA 124
HCD and HCM 49
HCD configuration reports
  viewing from HCM 279
HCD host problems 389
HCD messages 31
HCD profile, define 10
HCD trace facility, using the 26, 391
HCD-only tasks 331
HCM
  installing 3
  logical objects 54
  logon 25
  messages 33, 395
  overview 1
  physical objects 54
  starting 25

HCM *(continued)*
stopping 25
HCM agent
JCL for starting 411
HCM and HCD 49
HCM client 6
HCM configuration files
compare 281
HCM dispatcher 6
hcm file 50
HCM INI file 5
HCM trace data allocation 10
HCM trace facility, using the 10
hcmz 50
hcr file 50
hcrz 50
help
online 29
help information, incorrect 390
help menu choices summary 431
high level qualifier, specifying 63
highlighted objects pull down choice 33
highlighting objects, using configuration data 38
horizontal view of XMP processors 37
host access
IPv6 6
TCP/IP protocol 6
host communication
encrypt 19
host communication adapter 124
host communication requirements 4
host requirements 4

I

I/O Autoconfiguration
attended/unattended mode 224
configuration policy 228
discovered controllers 223
full mode discovery 222
mode of discovery 222
scope of discovery 222
starting the wizard 222
Summary page 227
working with proposed control units 224
working with proposed devices 225
I/O configuration
statements
importing 236
I/O devices
adding to an esoteric 257
assigning to units 143
creating them in a string 147
defining them to OS configs 148
deleting 149
editing 148
glossary definition 455
integrated 177
modifying 146
VOLSER reconciliation 97
I/O subsystem
copying 234
creating 234
editing 234
IBMDFLT 12
identifying problems 385
implied ports 231
import IODF 75
importing
I/O configuration statements 236
importing data
data format 333
exporting data 349
table format details 349
table summary 347
importing user data
stand-alone mode 345
include/exclude controllers 33
incorrect configuration diagram 388
incorrect help information 390
incorrect results 387
Index pull down choice 29
infrastructure connections
editing 212
ini file
ERRLOGDIR parameter 6
INI file 25
initial configuration, inspecting 65
inspecting the initial configuration 65
installation problems 5
installing
HCM in a LAN environment 6
installing HCM
overview 3
prerequisites 3
service levels 6
TCP/IP enablement 6
uninstalling HCM 23
upgrading HCM 5
integrated device
glossary definition 455
integrated I/O devices 177
interface labels
editing 136
interface type, CHPID 119
interfaces
changing number 135
internal control units
glossary definition 455
internal devices
glossary definition 455
internal structures
glossary definition 455
Internet Protocol
Version 6 6
inventory cables 208
IOCP or I/O configuration
statements
exporting 235
IODF
copying an existing one 71
copying/merging 331
creating a new configuration from 56
deleting an existing one 75
description 425
details 425
export 75

IODF *(continued)*
glossary definition 455
import 75
load / resync process 52
load and resync 417
loading 59
merging/copying 331
naming rules 66
preparing 57
production 72
resync / load process 52
resynchronization 53
upgrade from an earlier version 331
work 72
IODF and configuration file, data integrity between 52
IODF compare reports 279
IODF load
for MCF-enabled configuration 82
IODF naming convention 78
IODF objects
adding similar objects 331
IODFs
compare 279
IPv6 6
IQD channel
IEDN access 124
IQDX function 124
IQDX 123

J

JCL
for creating a user ID for UNIX System Services 9
for starting the HCM agent 411
HCM agent 433
jumper connections 215

K

keyboard
navigation 437
PF keys 437
shortcut keys 437
shortcuts 27
usage 27

L

labels
of adapters and interfaces 136
labels, cable, printing 211
LAN environment 6
LC flag 84, 95
LCSS 120, 456
line styles in a diagram 35
line styles in port matrix 159
line styles, configuration 38
link address format
changing 234
listing
consoles for an OS configuration 252
devices for a system-defined generic 260
devices for a user-defined generic 261
listing *(continued)*
devices for an esoteric 257
devices for an OS configuration 247
OS configurations to which devices are defined 250
system-defined generics 259
user-defined generics 260
load production IODF
with associated MCF 62
load work IODF
with associated MCF 61
loading an IODF
for MCF-enabled configuration 82
loading PDFs 379
loading the IODF 59
local area network 6
locate menu choices summary 428
locating a patchport 169
locating objects 43
log file 392
logical and physical objects in HCM 54
logical definition
glossary definition 456
logical definitions between controller and processor, editing 189
logical object
glossary definition 456
logical partition groups 229
logical partitions 114
logical without physical connection 103
logical-physical inconsistencies 42, 57
logon HCM 25
long hexagon / pentagon command 41
loopback connection for FICON switches 161
LP groups 229
LPARs 114

M

making a processor primary or alternate 415
making a production IODF 72
making a work IODF 72
managed CHPIDs 121
managing
cables 207
switch configurations 158
managing IPL attributes 332
managing physical cables 207
managing shadow processors during an HCM session 415
map proposed control units
AUTO_SS_DEVNUM_SCHEME 226
mapping behavior
for CHPIDs 101
mapping CHPIDs
in the copy wizards 100
master configuration file
introduction 54
maximum frame size 123
MCF
access rights 81
disabling 81
exporting 390
HCD version checking 85
HCM version checking 85
loading an IODF 82

MCF *(continued)*
opening disabled configuration 83
opening enabled configuration 83
physical mismatch resolution 84
scenarios 82
sharing 79
stand-alone processing 83
MCF/MUA
enabling 80
menu bar choices summary 425
menu choices summary
create menu 429
edit menu 426
file menu 425
help menu 431
locate menu 428
options menu 431
utilities menu 430
view menu 427
message list 31
message problems 387
messages
HCD 31
HCM 33
migrating to FICON channels 233
mismatch resolution
guidelines 87
modifying
cabinets 165
CHPIDs 117
control units 138
controllers 130
converters 150
crossbar switches 162
I/O devices 146
objects in your configuration 99
partitions 114
patchports 167
processors 106
strings 140
switches 154
units 145
mouse
usage 28
moving
switch's control units 140
moving port connections 231
MSI package 4
MUA
multi-user access 62
multi-user access
MUA 62
multiple access 6
multiple CHPIDs
connect to multiple control units 190
multiple connection targets
for controllers 184
for switches 195
multiple control units
connect to multiple CHPIDs 190
multiple exposure device
glossary definition 456
multiple users, concurrent use (of IODF) by 52
mutable connections 35
MVS, OS configurations 245

N

named views 36
naming convention of activity log 67
naming conventions 421
navigation
keyboard 437
new cable to assign to a connection 210
null device candidate list 183
number of adapters
changing 134
number of interfaces
changing 135
numbering scheme
AUTO_SS_DEVNUM_SCHEME 226

O

object information bar 42
objects
creating 45
deleting 46
editing 45
filtering 45
locating 43
ordering 46
selecting 43
one processor per partition machines 417
online help 29
online mode
glossary definition 457
open command 67
open special function 97
opening a configuration MCF-disabled by another user 83
opening an existing configuration 67
opening an existing configuration file and a new IODF 68
opening an existing configuration file and associated IODF 67
opening an MCF-enabled configuration 83
operating system configurations 245
operating system requirements 3
operation mode
attended/unattended mode 224
operation mode symbols 120
operation mode, CHPID 120
OPPP 417
OPPP machines, creating and deleting 417
options menu choices summary 431
order of controller interfaces
changing 135
ordering
control unit interfaces 165
controllers connected to a crossbar 165
crossbar channel interfaces 165
objects 46
objects connected to a CHPID 122
string interfaces 144
strings connected to the controller 132
OS configuration
D/R site OS configuration 246
OS configuration-EDT-esoteric hierarchy 245

OS configurations
adding a console 253
adding devices 248
adding devices to an esoteric within 251
copying 247
creating 246
deleting 247
editing 247
editing a console 254
listing consoles 252
listing devices for 247
listing to which devices are defined 250
MVS 245
removing a console 254
removing devices from 249
types 245
VM 245
OS groups 230
OSD channels 126
OSx channel types 124
out of memory 389
output format of reports 265
overview 1

P

page breaks 42
parallel connections 175
partition
copying 116
dynamic I/O 115
reserved for dynamic I/O 115
partitions
adding user data to 116
assigning to CHPIDs 126
creating 115
deleting 117
editing 116
glossary definition 457
granting I/O devices access to 115
modifying 114
operation mode 120
password 25
password phrase 25
password, expired 25
patchport naming format
predefined set 165
syntax of user-defined 170
user-defined 170
patchports
adding to fiber links 216
glossary definition 457
locating 169
modifying 167
removing from fiber links 216
PAV alias devices 182
PCHID
glossary definition 458
PCIe functions
working with 111
PDF
for a new controller 133
PDF *(continued)*
for controllers 133
PDFs
assigning to objects 56
changing 379
loading 379
overview 379
syntax 379
peer-to-peer remote copy
glossary definition 457
performance data
browsing 237
permanent connections 35
permanent connections from patchports
disconnecting 214
physical and logical objects in HCM 54
physical cables 207
physical channel identifier
glossary definition 458
physical connections 207
physical description file
for a new controller 133
for controllers 133
physical description files 379
physical mismatch resolution 83
physical mismatch resolution of an MCF-enabled configuration 84
physical objects, graphical representation 55
physical-logical inconsistencies 42, 57
policy for autoconfiguration 228
port connections, moving 231
port groups
glossary definition 458
port matrix
editing 159
moving ports 232
symbols 159
port matrix, generating default 158
port names
applying default 161
applying defaults 157
user specified 162
port number 25
port range 157
positioning a crossbar switch on a daisy-chain 219
positioning objects along daisy-chained connections 176
PPRC
glossary definition 457
scenario 325
PPRC connection
creating 192
deleting 193
editing 192
PPRC connections
importing and exporting 192
PPRC Connections 191
PPRCLINK 348
preference numbers 245
preferences 41
preparing the IODF 57
prerequisites 3
print cable labels 276
printing
cable labels 211

printing *(continued)*
configuration diagrams 263
diagram reports 264
text reports 264
printing reports/configuration diagrams, problems 388
printing the diagram 43
priority queuing 124
problem determination 385
problem handling 385
problem reporting data bases 393
problems
identifying 385
problems printing reports/configuration diagrams 388
problems with UIM provided help 391
processing configuration data 332
processing CONFIGxx members 332
processor
copy 109
XMP processor 106
processor view
horizontal view 37
vertical view 37
processors
adding user data to 110
creating 108
deleting 110
editing 110
glossary definition 458
modifying 106
product identifiers 386
production configuration file
glossary definition 458
production IODF
naming convention 66
production IODF, building 72
profile
HCD 10
IBMDFLT 12
PROFILE EXEC 14
PROFILE RSK 15
z/VM HCD 16
PROFILE EXEC 14
PROFILE RSK 15
prohibited switch port 160

Q

querying
supported hardware 331
UIMs 331

R

reconciliation, VOLSERs 97
record information 29
recording crossbar configurations 163
reducing the view 33
remote copy, peer-to-peer 325
removing
cable from a connection 210
cables 209
removing HCM 23
Repeat scenario 99
replacing converters 153
reporting problems 393
reporting problems to IBM 393
reports
footnotes, edit 265
generate 265
limitations 265
output format 265
scope 265
title, edit 265
reports (text and diagram)
generating and printing 264
output formats 264
reports and diagrams 263
requirements
hardware 3
host 4
software 3
workstation 3
reserved partition 115
restore named matrix set 38
restore named view 37
resync process / IODF load 52
resynchronization 53
Reusable Server Kernel 12, 15
reusing an existing cable for a connection 210
reversing an action 46
RMF DDS
problem determination 389
RSK 12, 15

S

SAID
glossary definition 461
SAMBA/SMB server 52
sample cable labels 276
sample diagram reports 266
sample reports 266
save as... 43
save command 78
save list as... 43
save named matrix set 38
save named view 37
saving views 36
saving your configuration 78
scaling printed diagrams 264
scenario
Repeat scenario 99
Upgrade scenario 99
scenarios 285
scope of discovery 222
screen elements 42
screen snapshot 263
scrolling the view 41
searching data bases 393
searching problem reporting data bases 393
secure communication
host 20
workstation 22
security
commands for setting 17, 391
segment
glossary definition 459

selecting a channel interface for a controller 183
selecting a CHPID to view details 121
selecting an existing string under a controller 178
selecting objects 43
selecting OS configurations from which to remove devices 251
selecting OS configurations to which to define devices 251
selection list 43
sending to IBM
  reader comments xxvii
serial connections 175
serial numbers 55–57
Server Time Protocol (STP) link 205
service levels
  installation 6
service virtual machine 11, 12
setting preferences 41
setting the size of a crossbar switch 164
shared access of production configuration file 52
sharing a configuration 79
sharing an MCF 79
shortcut keys 437
shortcuts 27
show page breaks 42
showing configurations 38
showing page breaks command 40
showing screen elements 42
SMP processor
  glossary definition 459
  term definition 106
snapshot, printing 263
software requirements 3
sorting selection list 43
spanned channel paths 120
spanned CHPID 120
special setup considerations 433
specifying a high level qualifier 63
specifying options 42
SSID
  glossary definition 460
stand-alone mode
  glossary definition 459
  invoking 95
stand-alone processing of an MCF-enabled configuration 83
standard printable labels 211
starting HCM 25
status bar 42
stopping HCM 25
storage group file 15
STP link 205
string interface from a device interface, disconnecting 183
string interfaces, changing the number of 145
string interfaces, ordering 144
string PDF Syntax 382
strings
  adding user data to 143
  assigning devices to units in 143
  changing the physical description of 144
  connecting 179
  creating 141
  deleting 145
  disconnecting 183
  editing 142
  glossary definition 459
strings *(continued)*
  making a custom connection 180
  making a physical connection 179
  making a standard connection 179
  modifying 140
subchannel set
  changing placement of devices 44, 149
  glossary definition 460
  placement of devices 149
subsystem
  glossary definition 460
subsystem identifier
  glossary definition 460
subsystems, grouping controllers into 137
summary of changes xxix
Summary page 227
support level, choosing 109
support of large IODFs 331
SVM 11, 12
switch
  glossary definition 460
  switch's control units 139
switch attributes
  editing 156
switch configuration
  line styles 38
  visualizing 38
switch configurations
  creating 158
  deleting 162
  editing 159
  managing 158
switch matrix, generating default 158
switch port
  allowed 160
  blocked 160
  prohibited 160
switch port attributes
  editing 157
switch port from a target interface, disconnecting 195
switch port names
  applying default 161
  user specified 162
switch ports
  glossary definition 460
switch ports in a configuration, editing 162
switch's control unit
  glossary definition 460
switch's control units
  deleting 140
  moving into a switch 140
switch's devices
  creating 147
  glossary definition 460
switches
  connecting 193
  creating 155
  deleting 158
  disconnecting 195
  editing 155
  modifying 154
switches, crossbar 162
switches, dynamic 154
switches, static 162

sysplex-wide activate 332
system adapter identifier
  glossary definition 461

T

table format
  PPRCLINK table 376
table format (export)
  CUCHPID table 377
  DEVCU table 377
  DEVIMAGE table 378
table format (import/export)
  CABINET table 366
  CABLE table 369
  CCI table 357
  CDI table 358
  CHPID table 352
  CONNECT table 373
  CONTRLLR table 355
  CONVERTR table 368
  CU table 359
  details 349
  DEVICE table 363
  DIR table 354
  general box port table 367
  general box table 366
  LOGLINK table 370
  PANEL table 367
  PART table 351
  PATCHPRT table 367
  PORT table 354
  PROC table 349
  SI table 362
  STRING table 360
  UNIT table 363
  XBAR table 364
  XBARCFG table 365
  XCI table 366
  XCUI table 366
tailoring your view 33
tasks, HCD-only 331
TCP/IP definitions
  IPv6 6
TCP/IP stacks 124, 126
text and diagram reports
  footnotes, edit 265
  title, edit 265
text detail 41
text reports
  samples of 266
timing-only link 205
title, text and diagram report 265
trace facility
  z/VM HCD TCP/IP dispatcher 391
trademarks 445
transferring partition configurations 331
transferring unused cables to inventory 208
trunk the two cabinets together 311
trunk the zone cabinet to the processor 312

U

UIM provided help, problems with 391
unattended mode 224
unattended node 93
undo 46
uninstalling HCM 23
units
  adding user data to 146
  assigning to devices 143
  creating 145
  deleting 146
  editing 145
  glossary definition 461
  modifying 145
UNIX System Services 4
unmanaged CHPIDs 121
unused cables to inventory , transferring 208
unused CTC connections 200
unused CTC control units 200
unused CTC CUs, devices, connections, deleting 200
unused CTC CUs, devices, connections, viewing 200
unused CTC devices 200
upgrade an IODF from an earlier version of HCD 331
Upgrade scenario 99
Upgrading
  HCM 69
  hcr 69
  production configuration files 69
  work configuration files 69
upgrading HCM 5
user data
  exporting 344
  exporting in stand-alone mode 345
  importing in stand-alone mode 345
user data fields 29
user ID mapping file 14
user interface
  ISPF 437
  TSO/E 437
user map file 14
user-defined general box port naming format 170
user-defined patchport naming format 170
using the HCD trace facility 391
using the keyboard 27
using the mouse 28
utilities
  aggregate CHPIDS 233
  browsing performance data 237
  changing link address format 234
  converting to FICON channels 233
  Copy Eligible Device Table 255
  moving port connections 231
utilities menu choices summary 430
utility
  Copy Channel Subsystem 113
  Copy Eligible Device Table 255
  Copy Esoteric Device Group 256
  Copy Operating System Configuration 247
  Copy partition 116
  Copy SMP Processor 109
  Copy XMP Processor 109
  create/copy/edit I/O subsystem 234

V

validated work IODF
    glossary definition 461
version checking of an MCF-enabled configuration 85
vertical view of XMP processors 37, 41
view
    saving 36
    scrolling 41
    tailoring 33
    zooming 41
view menu choices summary 427
viewing
    activity log 331
    cables 208
    connections 208
    CSS/OS compare reports 280
    CTC connection report 199
    HCD compare reports 279
    IODF compare reports 279
viewing HCD compare reports 279
viewing IODF compare reports 279
visual warning
    logical without physical connection 103
visual warnings 42, 53
visualized matrices 38
visualizing configurations 38
VM, OS configurations 245
VOLSER
    editing 97
    reconciliation 97
VOLSER reconciliation
    glossary definition 458

W

wizard
    Autoconfiguration 223
    Copy Channel Subsystem 113
    Copy Eligible Device Table 255
    Copy Esoteric Device Group 256
    Copy Operating System Configuration 247
    Copy partition 116
    Copy SMP Processor 109
    Copy XMP Processor 109
    create/copy I/O subsystem 131
    create/copy/edit I/O subsystem 234
work configuration file
    glossary definition 461
work IODF
    naming convention 66
work IODF, building 72
working with IODFs 71
workstation requirements 3
writing IOCDSs 332
WWPN Prediction Tool 242, 461

X

XMP processor
    glossary definition 461
    term definition 106
XMP processor *(continued)*
    vertical view 37, 41

Z

z/VM HCD Profile 16
z/VM HCD TCP/IP dispatcher
    commands 17, 391
z/VM HCD TCP/IP dispatcher for HCM
    commands 391
    service virtual machine 12
    start and stop 26
zoom in 41
zoom out 41
zooming the view 41



Product Number: 5650-ZOS

SC34-2670-50

