

z/OS



Hardware Configuration Definition User's Guide

Version 2 Release 2

Note

Before using this information and the product it supports, read the information in "Notices" on page 535.

This edition applies to Version 2 Release 2 of z/OS (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1994, 2016.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures ix

Tables xiii

About this document xv

Who this document is for xv

Related information xv

How to use this document xvi

How this document is organized xvi

How to read syntax diagrams xviii

 Symbols xviii

 Syntax items xviii

 Syntax examples xix

How to send your comments to IBM xxi

If you have a technical problem xxi

Summary of changes xxiii

Summary of changes for z/OS Version 2 Release 2 (V2R2) as updated September 2016 xxiii

Summary of changes for z/OS Version 2 Release 2 (V2R2) as updated March 2016 xxiii

Summary of changes for z/OS Version 2 Release 2 (V2R2) xxiv

Summary of changes for z/OS Version 2 Release 1 (V2R1) as updated February, 2015 xxiv

Summary of changes for z/OS Version 2 Release 1 (V2R1) xxv

 Enhancements of the I/O Autoconfiguration function xxv

 HMC-wide activate xxv

 Support of PCIe functions xxvi

 Validation enhancements xxvi

 Enhanced CF Channel Path Connectivity List OS group change action available for device groups xxvii

 HCD batch enhancements xxvii

 New profile options xxvii

 CHID Summary Report xxvii

 Verify a configuration by means of I/O Autoconfiguration (zDAC) xxvii

 Hardware support xxviii

Chapter 1. Hardware configuration definition - What is it? 1

What HCD is and how it differs from MVSCP and IOCP 1

 Definition of configuration data 1

 Validation of configuration data 3

 Reconfiguration of configuration data 4

What HCD offers you 4

How HCD works 5

 Objects managed in the IODF 6

 IODF used at IPL 6

 Relationship of data sets used by HCD 7

The environment in which HCD operates 8

 HCD and I/O Operations 9

 HCD and the coupling facility 9

 Support of z Systems clusters 11

 Support of the sysplex environment 11

Chapter 2. Migration 13

Migration overview 13

Migration roadmap 13

 Security migration 14

 Migration tasks 14

Chapter 3. How to set up, customize and start HCD. 15

Setting up HCD 15

 How to invoke HCD in dialog mode 16

 Tailoring the CLIST CBDCHCD 17

Starting and ending HCD 18

Defining an HCD profile 19

 Working with the HCD Profile Options dialog 20

 Keywords 22

Customizing HCD EXEC procedures 30

Chapter 4. How to work with I/O definition files (IODF) 31

IODF naming convention 31

 Work IODF 31

 Production IODF 32

 Associated data sets 32

Create or specify an IODF 32

 Multi-user access 33

 Sharing IODFs 34

Deciding on the number of IODFs 35

The master IODF concept 36

How HCD arranges devices into groups in an IODF 37

Change to another IODF 38

Change a production IODF / Create a work IODF based on a production IODF 38

View active IODF 38

Backup work or production IODFs 39

Maintain IODFs 39

 Delete an IODF 40

 Copy an IODF 40

 Change IODF attributes 41

 View an IODF 41

 Export an IODF 41

 Import an IODF 44

 Upgrade an IODF 44

 Work with configuration packages 46

Activity logging and change logging 50

 Rules for automatic activity logging 52

 Actions performed on IODFs and related activity and change log files 52

IODF release level compatibility 53

Chapter 5. How to use the dialog 55

Window layout	55
Working with lists	57
Numbered selection lists	57
Unnumbered single selection lists	57
Unnumbered multiple selection lists	58
Action lists	59
Message lists	62
Promptable fields	63
Commands and function keys	64
Getting help	64
Navigating through the dialog	65
Moving forward and backward within a task	65
Fast path	66
Filtering	69
Job statement information used in panels	72

Chapter 6. How to define, modify, or view a configuration 73

Creating new objects	73
Add	73
Add like	74
Repeat (copy).	74
Navigating through HCD.	74
Centralized navigation.	74
Hierarchical navigation	74
Graphical navigation	75
Navigation map	75
Suggested sequence to define a configuration	76
Working with operating system configurations	77
Defining operating system configurations	77
Changing operating system configurations	79
Repeating (copying) operating system configurations	79
Deleting operating system configurations	80
Working with EDTs.	80
Defining EDTs	80
Changing EDTs	81
Repeating (copying) EDTs	82
Deleting EDTs	82
Working with generics.	82
Working with esoteric groups	82
Defining esoteric groups	83
Changing esoteric groups.	83
Adding devices to esoterics	84
Repeating (copying) esoteric groups	84
Deleting esoteric groups	85
Working with processors	85
Defining processors.	85
Changing processors	87
Repeating (copying) processors.	89
Copying an SMP processor to an XMP channel subsystem	91
Priming processor data	92
Deleting processors.	92
Working with channel subsystems.	93
Defining channel subsystems	93
Repeating (copying) channel subsystems	94
Copying a channel subsystem to an SMP processor	96

Changing channel subsystems	96
Deleting channel subsystems	97
Working with partitions	97
Defining partitions	97
Changing partitions	99
Repeating (copying) partitions.	100
Transferring partition configurations.	100
Deleting partitions.	102
Working with PCIe functions	103
Defining PCIe functions	103
Changing PCIe functions	107
Deleting PCIe functions	108
Working with channel paths	108
Using Multiple Image Facility	108
More about spanned channel paths in multiple LCSSs	108
Defining channel paths	109
Establishing coupling facility channel path connections	117
Defining Server Time Protocol (STP) links	120
Disconnecting coupling facility channel path connections	120
Changing channel paths.	120
Aggregating channel paths	124
Deleting channel paths	126
Working with control units	126
Defining control units	126
Changing control units	130
Changing control unit attachment parameters for multiple processors	132
Disconnecting control units from a processor	132
Priming control unit data	133
Deleting control units	133
Working with devices	134
Defining devices	134
Changing devices	141
Disconnecting devices from an operating system	147
Showing or hiding parameter/feature definitions of devices.	148
Priming device data	149
Deleting devices	150
Working with operating system consoles	150
Changing operating system consoles	151
Deleting operating system consoles	151
Working on IODFs enabled for multi-user access	151
Simultaneously updating and viewing an IODF	151
Concurrently updating an IODF	152
Immediately reflecting changes during concurrent updates	153
Viewing information	154
Viewing object definitions	154
Viewing coupling facility information	155
Viewing CTC connections	156

Chapter 7. How to work with switches 163

Possibilities of switch connections	163
Advantages of switch definitions with HCD	165
Defining switches	166
Working with switches	166
Defining switch characteristics.	166
Changing switch data	169

Priming switch data	169
Moving ports	170
Deleting switches	171
Working with connections to switches	172
Defining connections to switches	172
Priming switch port data	174
Additional actions on the port list	175
Working with switch configurations	177
Defining switch configuration data	177
Changing the switch configuration ID	179
Generating a switch matrix	179
Deleting switch configurations	181
Migrating existing switch configurations	181
Prerequisites	181
Migration steps	181
Activating switch configuration data	183
Prerequisites	183
Activation steps	183
Saving switch configuration data	184

Chapter 8. How to work with I/O

Autoconfiguration 187

How to define autoconfiguration policies	188
How to set keywords for autoconfiguration policies	188
How to change autoconfiguration policies during the autoconfiguration process	192
How to define logical partition groups for autoconfiguration	193
How to define OS groups for autoconfiguration	194
How to perform automatic I/O configuration	196
How to apply updates to the autoconfiguration proposals	199
How to work in unattended mode	203
Prerequisites and operational considerations for using I/O Autoconfiguration	203

Chapter 9. How to activate or process configuration data 207

Build a production IODF	208
Build an IOCDS	212
Build z Systems cluster IOCDSs	215
Manage z Systems cluster IPL attributes	218
Build an IOCP input data set	219
How to build an IOCP data set	219
IOCP enhancements	221
IOCP input data sets using extended migration	221
Using the IOCP data set as input for the CHPID Mapping Tool	223
How to interact with the CHPID Mapping Tool	224
Process overview	224
How to insert or update PCHIDs	224
Create JES3 initialization stream checker data	226
Build I/O configuration data	226
Verify an I/O configuration	227
Verifying a configuration against the local system	228
Verifying a configuration against a system in the sysplex	228
The I/O path list	229

Activate a configuration dynamically	230
View active configuration	230
How to activate if hardware and software changes are allowed	231
How to activate if software-only changes are allowed	234
Activate a configuration sysplex-wide	234
Displaying active sysplex members	235
Activate software configuration changes only	236
Activate software and hardware configuration changes	238
Switch IOCDS for the next POR	239
Activate a configuration HMC-wide	239
Available configuration activation actions	242
Establishing connectivity to remote systems	245
Prerequisites for working with CPC images	246
Build a CONFIGxx member	247
Process the Display M=CONFIG(xx) command	249
Switch IOCDS for next POR	249
Switch IOCDS for processor without SNA address	249
Switch IOCDS for processors in a z Systems cluster with SNA address defined	250
Switch IOCDS for systems in a sysplex	250
Specify an IODF for IPL	251
IODF processing at IPL	251

Chapter 10. How to print and compare configuration data 253

Print configuration reports	253
Channel Subsystem Report	253
Switch Report	254
Operating System Report	254
CTC Connection Report	254
I/O Path Report	254
Supported Hardware Report	255
I/O Definition Reference	255
How to print a textual report	255
Create or view graphical configuration reports	257
Prerequisites	257
How to create a graphical configuration report	258
Printing the output	259
Viewing the output	260
How to print list panels	263
HCD compare functions	265
Compare IODFs	265
Compare CSS / operating system views	269
View and print the HCD activity log	270

Chapter 11. How to query supported hardware and installed UIMs 271

Query supported processors	271
Query supported switches	273
Query supported control units	274
Query supported devices	275
Query installed UIMs	276

Chapter 12. How to migrate existing input data sets 279

Migration sequence	279
------------------------------	-----

LPAR considerations	279
Preparing your input data sets for migration	280
Data requiring attention	280
Preparing additional input data sets for migration.	282
Migrating input data sets using the HCD dialog	287
Step 1: Specify the work IODF.	287
Step 2: Migrate the input data sets	287
Step 3: Analyze errors and correct the input data	289
Step 4: Update configuration data	289
Migrating input data sets using the batch utility	290
Step 1: Create the work IODF	290
Step 2: Migrate input data set	290
Step 3: Analyze errors and correct the input data	290
Step 4: Build production IODF	290
Changing I/O configurations by editing data sets	291
Additional parameters and statements	292
Updating parts of a configuration by migrating input data sets	313
Possible actions on objects using the incremental update	314
How to invoke the incremental update	317
Resolving migration errors	322
Errors detected during assembly process	322
Errors detected during validation process	324
Insufficient data set sizes	325

Chapter 13. How to invoke HCD batch utility functions 327

Input parameter string	328
Start the dialog.	328
Initialize IODF	329
Upgrade IODF	330
Migrate I/O configuration statements	331
Build a Production IODF	334
Build a work IODF from a production IODF	335
Build an IOCDS or an IOCP input data set	335
Activate a production IODF	338
Build an HCPRIO input data set	338
Build I/O configuration data	339
Copy IODF	342
Print configuration reports	343
Create a graphical configuration report.	346
Compare IODFs or CSS/OS Reports.	348
Import an IODF	352
Export an IODF	353
Batch IODF copy example	355
List of standard DD names	357
List of alternate DD names	358

Chapter 14. Security and other considerations 359

Security-related considerations	359
Defining RACF profiles	359
Giving users access authority	360
Access to HWL.* profiles.	362
How to set up PassTickets for working with CPC images on z/OS	362

How to set up PassTickets for working with CPC images on z/VM	364
Providing additional security for devices	364
Catalog considerations	365
Data sets cataloged with an esoteric device group name	365
SMS-related considerations	366
ISPF-related considerations	367
z/VM-related considerations	367

Chapter 15. How to provide LDAP support for HCD 369

Introduction to LDAP	370
HCD LDAP backend: Structure and mode of operation.	371
Plugging the HCD LDAP backend into the IBM Tivoli Directory Server for z/OS	372
Performing the setup.	373
Customizing the HCD LDAP backend	377
Configuration file parameters	377
LDAP debug level.	380
Message translation	380
Accessing IODF information	381
The IODF directory information tree.	381
Performing LDAP requests on IODFs	382
Operational behavior	382
Restrictions for search and update requests	383
Transactions	387
Prerequisites and method of functioning	387
How to initiate, extend and close a transaction	389

Appendix A. How to navigate through the dialog 393

Appendix B. Configuration reports 399

Textual configuration reports	399
Channel subsystem reports	399
Switch reports	414
Operating System reports	416
CTC Connection Report	422
I/O Path Report	423
Supported hardware report.	425
I/O Definition Reference	451
Graphical configuration reports	454
LCU Report	454
IODF compare reports	454
Processor Compare Report	455
Channel Subsystem Compare Report	456
PCIe Function Compare Report	456
Partition Compare Report	458
Channel path compare report	459
Control Unit Attachment Compare Report.	460
Device Attachment Compare Report.	461
Control Unit Compare Report	461
Device Compare Report	462
Switch Compare Report	463
Switch Detail Compare Report	463
Switch Configuration Compare Report	464
Switch Configuration Detail Compare Report	464
Esoteric Compare Report	465

Operating System Compare Report	466
OS Device Compare Report	467
OS Console Compare Report	467
CSS / OS Device Compare Report	468

Appendix C. Problem determination for HCD 469

Identifying problems	469
HCD abnormal termination	470
Error during IPL (Wait State Codes)	472
Problems with panels and function key assignment	473
Problems with help information provided by HCD	474
Problems with output of HCD textual reports	475
Problems with output of HCD graphical reports	475
Problems during initialization of HCD	477
Problems with UIMs	478
HCD internal problems	479
Problems with the Transmit Configuration Package action	480
LDAP problem determination	484
Diagnostic information and tools	487
HCD messages	487
HCD trace facility	488
IODF dump	494
Repair an IODF	495
MVS dumps and traces	496
IPCS reports	496
Searching problem reporting data bases and reporting problems	496
Sending an IODF to a different location	496

Appendix D. HCD object management services 499

How to invoke the HOM services	499
Data input and output areas	500
Request block (HRB)	501
Functions	502
Example	503

Return codes	504
Reason codes	504

Appendix E. Establishing the host communication 505

Setting up TCP/IP definitions for z/OS	505
Controlling access to HCD services	507
Skeleton used to start the HCD agent	507
Starting the HCD dispatcher as a started task	508
Starting the HCD dispatcher as a batch job	509
Stopping the HCD dispatcher	509
Define an HCD profile	509
Allocate the HCD trace data set for remote HCD sessions	509
Verifying TCP/IP host communication	510
TCP/IP problem determination	510

Appendix F. IODF data model 511

IODF object class definitions within LDAP	512
IODF attribute table	521

Appendix G. Accessibility 531

Accessibility features	531
Consult assistive technologies	531
Keyboard navigation of the user interface	531
Dotted decimal syntax diagrams	531

Notices 535

Policy for unsupported hardware	536
Minimum supported hardware	537

Trademarks 539

Programming interface information 541

Glossary 543

Index 551

Figures

1. Multiple data set configuration definition without HCD	2	51. Add Channel Path	110
2. Single data set configuration definition with HCD	3	52. Channel Path/Partition Matrix	112
3. Configuration definition with HCD	6	53. CF Channel Path Connectivity List	117
4. Objects managed by HCD	6	54. Connect to CF Channel Path	118
5. Building a production IODF and an IOCDS	7	55. Add CF Control Unit and Devices	119
6. Relationship of data sets used by HCD.	8	56. Select Processor / Control Unit (1)	128
7. Processor configuration with coupling facility implemented	10	57. Select Processor / Control Unit (2)	128
8. Sample ISPF Master Application Menu	17	58. Change Channel Path Link Addresses	130
9. HCD <i>Primary Task Selection</i> panel	18	59. I/O Device List with device groups	135
10. Example of an HCD Profile	29	60. I/O Device List with single devices	135
11. Create Work I/O Definition File.	33	61. Add Device	136
12. IODF Distribution/Merge Process	37	62. Device / Processor Definition	137
13. Maintain I/O Definition Files	39	63. Define Device / Processor	138
14. Create New I/O Definition File	40	64. Define Device Candidate List	139
15. View IODF information	41	65. Specify Subchannel Set ID	140
16. Define JCL for IODF Import	42	66. Specify Subchannel Set ID	146
17. Specify Target IODF and User Password	43	67. Eligible Channel Subsystems	146
18. Upgrade an IODF	44	68. Eligible Operating System Configurations	147
19. Create Work I/O Definition File.	46	69. CTC/CNC connection established using a dynamic connection	156
20. Sample Configuration Package List — left panel.	46	70. CTC/CNC connection established using shared channels.	157
21. Add Configuration Package	47	71. CTC connection established using point-to-point	158
22. Transmit Configuration Package	49	72. Configuration with one switch.	164
23. Example of an Activity Log without automatic logging	51	73. Configuration with two switches (1)	164
24. View Activity Log	52	74. Configuration with two switches (2)	165
25. Example of a window layout.	55	75. Configuration with two cascading FICON switches	165
26. Layout of a data entry dialog	56	76. Switch List (left part)	167
27. <i>Primary Task Selection</i> panel	57	77. Switch List (right part)	167
28. Example of a single selection list	58	78. Add Switch	168
29. Example of a Multiple Selection List	59	79. Change Switch Definition	169
30. Action list with a context menu.	60	80. Confirm Priming Switch Data List	170
31. Action list with action codes	61	81. Move Ports to a Target Switch	170
32. Message List	62	82. Confirm Delete Switch	172
33. Explanation Message	62	83. Port List	173
34. Example of a Prompt Selection Panel	63	84. Profile Options and Policies.	188
35. Using the Goto Action Bar Choice	66	85. Autoconfiguration Policies	189
36. Operating System Configuration List as example of a Selection List	67	86. Discovery and Autoconfiguration Options	197
37. Filter example.	70	87. Discovered New or Changed Controller List	199
38. Filter I/O Device List	71	88. Proposed Control Unit List	200
39. Reduced I/O Device List (using a filter)	71	89. Proposed Control Unit / Device List	202
40. Navigation map	76	90. Activate or Process Configuration Data	209
41. Operating System Configuration List	78	91. Build Production I/O Definition File	210
42. Add a processor	86	92. Confirm Production IODF Name	211
43. Available support levels	86	93. Confirm Delete I/O Definition File	211
44. CTC Connection Update List.	95	94. Define Descriptor Fields	211
45. Confirm Delete Partition.	103	95. IOCDS List for locally building the IOCDS	212
46. PCIe Function List	104	96. z Systems Cluster List	215
47. PCIe Function List	105	97. IOCDS List invoked from the z Systems Cluster List	216
48. Add PCIe Function	105	98. Build IOCP Input Data Set	220
49. Add/Modify Physical Network IDs	106	99. Example of an input data set for migration enhancements	223
50. Define Access List	107	100. Identify System I/O Configuration	228

101. Activate New Configuration: Hardware and Software Changes	232	143. Message List containing an Assembler message	323
102. Activate Sysplex Member List	235	144. Example: HCD Migration Log	324
103. Activate Software Configuration Only	236	145. Build IOCDS job	337
104. Message panel	237	146. Build IOCP input data set	338
105. View Configuration Status	237	147. Activate IODF job	338
106. Activate Hardware and Software Configuration	238	148. Recommended IODF catalog structure	365
107. CPC Image List	241	149. Define a VM operating system	367
108. CPC Image List scrolled to the right	242	150. Define Device Parameters/Features for VM Device	368
109. Process System Command	244	151. Sample DIT portion of the HCD LDAP backend	371
110. Process System Command - returned messages	245	152. Primary Task Selection panel	393
111. Restrict Ports Eligible for Dynamic CHPID Management.	247	153. HCD - Edit profile options and policies - Option 0	393
112. Build CONFIGxx Member	248	154. HCD - Define, Modify, or View Configuration Data - Option 1	394
113. Confirm Update CONFIGxx Member	248	155. HCD - Define Operating System - Option 1.1	394
114. Process Display M=CONFIG(xx) Command	249	156. HCD - Define Switch - Option 1.2.	395
115. Print Configuration Reports.	255	157. HCD - Define Processor and Channel Path - Option 1.3	395
116. Limit Reports	256	158. HCD - Define Control Unit - Option 1.4	395
117. Viewing a sample report	260	159. HCD - Define I/O Device - Option 1.5	396
118. Example of a printed list.	265	160. HCD - Primary Tasks - Options 2 - 7	396
119. Compare IODFs.	266	161. HCD - Generic Action Bar Options	397
120. Limit Processor Compare Reports.	267	162. Processor Summary Report	400
121. Limit Switch Compare Reports.	267	163. Channel Subsystem Summary Report	400
122. Limit Operating System Compare Reports	268	164. PCIe Function Summary Report	401
123. Select Print Options (for Compare IODFs only)	268	165. CHID Summary Report	402
124. Select Print Options (for CSS/OS Compare only)	269	166. Partition Report.	403
125. Query supported processors	271	167. IOCDS Report	404
126. IOCP Migration.	283	168. Channel Path Summary Report	405
127. IOCP Migration.	283	169. Channel Path Detail Report	407
128. IOCP Migration.	284	170. CF Channel Path Connectivity Report	408
129. IOCP Migration.	284	171. Control Unit Summary Report	409
130. IOCP Migration.	285	172. Control Unit Detail Report	411
131. IOCP Migration.	285	173. Device Summary Report	412
132. MVSCP Migration	286	174. Device Detail Report	414
133. MVSCP Migration	286	175. Switch Summary Report	415
134. MVSCP Migration	286	176. Switch Detail Report	415
135. Migrate IOCP / MVSCP / HCPRIO Data	287	177. Switch Configuration Summary Report	415
136. Partial Migration of an IOCP Input Data Set. A new partition is added. Control units and devices are mapped.	318	178. Switch Configuration Detail Report	416
137. Partial migration of an IOCP input data set. The whole logical control unit (control unit 100 and 200 and connected devices) are replaced by control unit 100 and its connected devices.	319	179. Operating System Summary Report	416
138. Partial migration of an IOCP input data set. CHPIDs are defined as in the IOCP input data set.	319	180. MVS Device Report	417
139. Partial migration of an IOCP input data set. Control unit defined for two processors is migrated to the first processor.	320	181. MVS Device Detail Report	418
140. Partial migration of an IOCP input data set. Control unit defined for two processors is migrated to the second processor.	320	182. Eligible Device Table Report	419
141. Partial migration of an MVSCP input data set	321	183. NIP Console Report	420
142. Message List	322	184. VM Device Report	420
		185. VM Device Detail Report	421
		186. VM Console Report	421
		187. CTC Connection Report	422
		188. Sample of Diagnostic Messages coming with the CTC connection report	423
		189. Example and Legend of an I/O Path Report (Part 1 of 2)	424
		190. Example and Legend of an I/O Path Report (Part 2 of 2)	425
		191. Supported Hardware Report (Part 1 of 30)	427
		192. Supported Hardware Report (Part 2 of 30)	428
		193. Supported Hardware Report (Part 3 of 30)	429
		194. Supported Hardware Report (Part 4 of 30)	430

195. Supported Hardware Report (Part 5 of 30)	431	224. I/O Definition Reference (Part 4 of 5)	453
196. Supported Hardware Report (Part 6 of 30)	432	225. I/O Definition Reference (Part 5 of 5)	454
197. Supported Hardware Report (Part 7 of 30)	433	226. Processor Compare Report	455
198. Supported Hardware Report (Part 8 of 30)	433	227. Channel Subsystem Compare Report	456
199. Supported Hardware Report (Part 9 of 30)	434	228. PCIe Function Compare Report	457
200. Supported Hardware Report (Part 10 of 30)	435	229. Partition Compare Report	458
201. Supported Hardware Report (Part 11 of 30)	436	230. Channel Path Compare Report	459
202. Supported Hardware Report (Part 12 of 30)	436	231. Control Unit Attachment Compare Report	460
203. Supported Hardware Report (Part 13 of 30)	437	232. Device Attachment Compare Report	461
204. Supported Hardware Report (Part 14 of 30)	438	233. Control Unit Compare Report	461
205. Supported Hardware Report (Part 15 of 30)	439	234. Device Compare Report	462
206. Supported Hardware Report (Part 16 of 30)	440	235. Switch Compare Report	463
207. Supported Hardware Report (Part 17 of 30)	441	236. Switch Detail Compare Report	463
208. Supported Hardware Report (Part 18 of 30)	442	237. Switch Configuration Compare Report	464
209. Supported Hardware Report (Part 19 of 30)	443	238. Switch Configuration Detail Compare Report	464
210. Supported Hardware Report (Part 20 of 30)	443	239. Esoteric Compare Report.	465
211. Supported Hardware Report (Part 21 of 30)	444	240. Operating System Compare Report	466
212. Supported Hardware Report (Part 22 of 30)	445	241. OS Device Compare Report	467
213. Supported Hardware Report (Part 23 of 30)	446	242. OS Console Compare Report	467
214. Supported Hardware Report (Part 24 of 30)	447	243. CSS / OS Device Compare	468
215. Supported Hardware Report (Part 25 of 30)	447	244. Example: Trace output	490
216. Supported Hardware Report (Part 26 of 30)	448	245. Example: Trace output in case of Abend	491
217. Supported Hardware Report (Part 27 of 30)	449	246. Example: JCL stream for producing an IODF	
218. Supported Hardware Report (Part 28 of 30)	450	dump	495
219. Supported Hardware Report (Part 29 of 30)	450	247. Relationship between HCD client, HCD	
220. Supported Hardware Report (Part 30 of 30)	451	dispatcher, and HCD agent	505
221. I/O Definition Reference (Part 1 of 5)	451	248. Sample JCL for creating a user ID for UNIX	
222. I/O Definition Reference (Part 2 of 5)	452	System Services	509
223. I/O Definition Reference (Part 3 of 5)	453	249. Sample job for trace data set allocation	510

Tables

1. Syntax examples	xix	28. APF authorized Load Module Libraries	376
2. Batch Jobs Used by the HCD Dialog	30	29. Description of the columns in the Control Unit Detail Report	411
3. Size considerations when upgrading a back-level IODF	45	30. Symptoms of system problems.	469
4. Coexistence Considerations for IODFs.	53	31. Search Argument	471
5. Online Help Information	64	32. Problem Data	471
6. Result of the Transfer (Move) Partition Configs Action	101	33. Search Argument	472
7. Operating System Syntax	295	34. Problem Data	472
8. NIP consoles Syntax	295	35. Search Argument	472
9. EDT Syntax	296	36. Problem Data	473
10. Esoteric Syntax	296	37. Search Argument	473
11. Generic Syntax	297	38. Problem Data	473
12. Switch Syntax	298	39. Search Argument	474
13. Port Syntax	299	40. Search Argument	474
14. Switch Configuration Syntax	299	41. Problem Data	475
15. Port Configuration Syntax	300	42. Problem Data	475
16. Processor Syntax	301	43. Problem Data	477
17. Channel subsystem and partition Syntax	302	44. Search Argument	478
18. PCIe user defined identifier syntax	304	45. Problem Data	478
19. PCIe Function Syntax	304	46. Search Argument	479
20. Channel Path Syntax	307	47. Problem Data	479
21. Control Unit Syntax	309	48. Search Argument	479
22. Device Syntax	311	49. Problem Data	480
23. Actions on IODF Objects.	314	50. Problem Data	484
24. Standard DD names Used by HCD	357	51. Where to find HCD messages	488
25. CBD.CPC.IPLPARM access authority and HCD IPL attribute management functions	361	52. Used registers and passed parameters	499
26. CBD.CPC.IOCDS access authority and HCD IOCDS management functions	361	53. Summary of Request Block Names and Related Constants	501
27. Load Module Libraries to be defined to program control.	375	54. Functions provided by the HOM services	502

About this document

This document explains how to use Hardware Configuration Definition (HCD) to accomplish the following:

- Define new hardware configurations
- Modify existing hardware configurations
- View existing hardware configurations
- Activate configurations
- Query supported hardware
- Maintain IODFs
- Compare two IODFs
- Compare an IODF with an actual configuration
- Print reports of configurations
- Create graphical reports of a configuration
- Migrate existing configuration data

Note

Unless otherwise noted, the term **MVS** as used in this document refers collectively to the older MVS operating system as well as to its successors OS/390 and z/OS, in which MVS is one of the basic components. For purposes of this document, MVS used alone and without reference to a specific release is to be understood as the generic operating system type supported by HCD.

Who this document is for

This document is for the person who is responsible for defining the hardware configuration for a z/OS system. It is assumed that:

- The person has a basic knowledge of z/OS, and hardware configuration.
- Configuration planning has been completed *before* HCD is used to enter definition data. For information on configuration planning, refer to *z/OS HCD Planning*.

For the person responsible for problem determination, this document also explains what to do if a problem arises with HCD.

This information is for system programmers and system operators.

Related information

Please see the *z/OS V2R2 Information Roadmap* for an overview of the documentation associated with z/OS.

To view, search, and print z/OS publications, go to the z/OS Internet Library at <http://www.ibm.com/systems/z/os/zos/bkserv/>.

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

Also visit our HCD home page

<http://www.ibm.com/systems/z/os/zos/features/hcm/>

which provides information concerning product updates, newsletters, conferences, and more.

How to use this document

Before you start to use HCD, you should read

- Chapter 1, “Hardware configuration definition - What is it?,” on page 1 and
- Chapter 5, “How to use the dialog,” on page 55.

These topics provide information about the general concepts and facilities of HCD.

If you want to use HCD to define a new configuration, you should read

- Chapter 5, “How to use the dialog,” on page 55,
- Chapter 6, “How to define, modify, or view a configuration,” on page 73, and
- Chapter 7, “How to work with switches,” on page 163.

Chapter 5, “How to use the dialog,” on page 55 explains how to use the HCD panels, get online help information, enter data and select items such as tasks, objects and actions. Chapter 6, “How to define, modify, or view a configuration,” on page 73 and Chapter 7, “How to work with switches,” on page 163 explain how to define operating system (OS) configurations, processors, control units, I/O devices, and switches.

How this document is organized

Chapter 1, “Hardware configuration definition - What is it?,” on page 1 explains how HCD fits into the context of hardware configurations and systems management. It also explains the environment in which HCD is used.

Chapter 2, “Migration,” on page 13 discusses how to move from a previous HCD release to HCD under z/OS. It also refers to other sections in this document dealing with migration and conversion tasks.

Chapter 3, “How to set up, customize and start HCD,” on page 15 provides information on how to install, customize, and start HCD, and how to set up an HCD installation for the first time in z/OS.

Chapter 4, “How to work with I/O definition files (IODF),” on page 31 explains how to work with I/O definition files (IODFs), for example, creating, changing, viewing, and deleting them. It also explains how to use configuration packages to create subset IODFs for distribution.

Chapter 5, “How to use the dialog,” on page 55 explains the general facilities of the HCD dialog, that is panels, online help, navigation, making selections, and entering data.

Chapter 6, “How to define, modify, or view a configuration,” on page 73 explains the navigation through the HCD dialog and how to define, change, copy, delete, and view operating system configurations, processors, control units, and devices. It also explains how to prime processor, control unit, and device data.

Chapter 7, “How to work with switches,” on page 163 includes basic information about switches and explains how to define, change, prime, and delete switches, how to define and prime switch connections (ports), and how to work with switch configurations. It also describes how to migrate, activate, and save switch configuration data.

Chapter 8, “How to work with I/O Autoconfiguration,” on page 187 explains how to perform automatic discovery and definition of switched FICON connected DASDs and tape control units and devices. Also, it describes how to specify options and policies how HCD should process the automatic I/O configuration.

Chapter 9, “How to activate or process configuration data,” on page 207 explains how to make a configuration available for use by the system. It further explains how to compare a configuration defined in an IODF with the configuration sensed on the system. It includes information on activating a configuration dynamically, activating a configuration sysplex-wide, and on remote IOCDs management functions.

Chapter 10, “How to print and compare configuration data,” on page 253 explains how to build textual and graphical reports about channel subsystem, switch, and operating system configuration, I/O paths, and CTC definitions and how to compare IODFs. It also explains how to print the data that is currently displayed on a list panel.

Chapter 11, “How to query supported hardware and installed UIMs,” on page 271 explains how to use HCD to view system data.

Chapter 12, “How to migrate existing input data sets,” on page 279 contains information for migrating existing IOCP/MVSCP/HCPRIO definitions and explains the steps in the migration process.

Chapter 13, “How to invoke HCD batch utility functions,” on page 327 describes the HCD programming interface.

Chapter 14, “Security and other considerations,” on page 359 provides information on various HCD-related topics.

Chapter 15, “How to provide LDAP support for HCD,” on page 369 explains the provision for retrieving and updating IODF data via LDAP protocol.

Appendix A, “How to navigate through the dialog,” on page 393 illustrates the flow from the HCD main panel options and the various actions that can be taken from each option.

Appendix B, “Configuration reports,” on page 399 contains examples of the various reports that can be printed by using HCD.

Appendix C, “Problem determination for HCD,” on page 469 explains what to do if problems occur with HCD.

Appendix D, “HCD object management services,” on page 499 explains how to use the HCD application programming interface to retrieve configuration data, such as switch data, device type, or control unit type, from the IODF.

Appendix E, “Establishing the host communication,” on page 505 describes how to set up TCP/IP definitions for z/OS target systems as a prerequisite for working with CPC images, or how to link the host and the workstation if you want to work with HCM.

Appendix F, “IODF data model,” on page 511 describes the IODF data model in terms of object classes and attribute definitions as used by the HCD LDAP backend.

How to read syntax diagrams

This section describes how to read syntax diagrams. It defines syntax diagram symbols, items that may be contained within the diagrams (keywords, variables, delimiters, operators, fragment references, operands) and provides syntax examples that contain these items.

Syntax diagrams pictorially display the order and parts (options and arguments) that comprise a command statement. They are read from left to right and from top to bottom, following the main path of the horizontal line.

For users accessing the Information Center using a screen reader, syntax diagrams are provided in dotted decimal format.

Symbols

The following symbols may be displayed in syntax diagrams:

Symbol	Definition
--------	------------

- | | |
|-----|--|
| ▶— | Indicates the beginning of the syntax diagram. |
| —▶ | Indicates that the syntax diagram is continued to the next line. |
| ▶— | Indicates that the syntax is continued from the previous line. |
| —▶◀ | Indicates the end of the syntax diagram. |

Syntax items

Syntax diagrams contain many different items. Syntax items include:

- Keywords - a command name or any other literal information.
- Variables - variables are italicized, appear in lowercase, and represent the name of values you can supply.
- Delimiters - delimiters indicate the start or end of keywords, variables, or operators. For example, a left parenthesis is a delimiter.
- Operators - operators include add (+), subtract (-), multiply (*), divide (/), equal (=), and other mathematical operations that may need to be performed.
- Fragment references - a part of a syntax diagram, separated from the diagram to show greater detail.
- Separators - a separator separates keywords, variables or operators. For example, a comma (,) is a separator.

Note: If a syntax diagram shows a character that is not alphanumeric (for example, parentheses, periods, commas, equal signs, a blank space), enter the character as part of the syntax.

Keywords, variables, and operators may be displayed as required, optional, or default. Fragments, separators, and delimiters may be displayed as required or optional.

Item type

Definition

Required

Required items are displayed on the main path of the horizontal line.

Optional

Optional items are displayed below the main path of the horizontal line.

Default

Default items are displayed above the main path of the horizontal line.

Syntax examples

The following table provides syntax examples.

Table 1. Syntax examples

Item	Syntax example
Required item.	
Required items appear on the main path of the horizontal line. You must specify these items.	
Required choice.	
A required choice (two or more items) appears in a vertical stack on the main path of the horizontal line. You must choose one of the items in the stack.	
Optional item.	
Optional items appear below the main path of the horizontal line.	
Optional choice.	
An optional choice (two or more items) appears in a vertical stack below the main path of the horizontal line. You may choose one of the items in the stack.	
Default.	
Default items appear above the main path of the horizontal line. The remaining items (required or optional) appear on (required) or below (optional) the main path of the horizontal line. The following example displays a default with optional items.	
Variable.	
Variables appear in lowercase italics. They represent names or values.	

Table 1. Syntax examples (continued)

Item	Syntax example
Repeatable item.	
An arrow returning to the left above the main path of the horizontal line indicates an item that can be repeated.	
A character within the arrow means you must separate repeated items with that character.	
An arrow returning to the left above a group of repeatable items indicates that one of the items can be selected, or a single item can be repeated.	
Fragment.	
The fragment symbol indicates that a labelled group is described below the main syntax diagram. Syntax is occasionally broken into fragments if the inclusion of the fragment would overly complicate the main syntax diagram.	

How to send your comments to IBM

We appreciate your input on this documentation. Please provide us with any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

Use one of the following methods to send your comments:

Important: If your comment regards a technical problem, see instead “If you have a technical problem.”

- Send an email to mhvrcfs@us.ibm.com.
- Send an email from the <http://www.ibm.com/systems/z/os/zos/webqs.html>.

Include the following information:

- Your name and address
- Your email address
- Your phone or fax number
- The publication title and order number:
 - z/OS V2R2 HCD User's Guide
 - SC34-2669-04
- The topic and page number or URL 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 right 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

Do not use the feedback methods that are listed for sending comments. Instead, take one or more of the following actions:

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

Summary of changes

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

Note: To see a description of new functionality of the current release, select option **9. What's new in this release** from the HCD *Primary Task Selection* panel. Here you may also find information about SPEs that are delivered after the completion of this document.

Also, visit our homepage at <http://www.ibm.com/systems/z/os/zos/features/hcm/>

Summary of changes for z/OS Version 2 Release 2 (V2R2) as updated September 2016

The following changes are made to z/OS Version 2 Release 2 (V2R2).

Hardware Support

PCIe UID support:

To provide the ability to enable/disable UID-checking for a logical partition a new attribute (UID) for PCIe functions is introduced and works in combination with the new partition attribute (uniqueness flag). A function has a unique UID if there is no other PCIe function accessible by the same partition that has the same UID defined.

Processor support:

HCD supports the IBM z13 processor family with UID support:

- processor types 2964-N30, -N63, -N96, -NC6, -NE1 (new support level H160930)
- processor types 2964-L30, L63, -L96, -LC6, -LE1 (supported level H160930)
- processor types 2965-N10 and N20 (support level H160930)
- processor types 2965-L10 and L20 (support level H160930)

Summary of changes for z/OS Version 2 Release 2 (V2R2) as updated March 2016

The following changes are made to z/OS Version 2 Release 2 (V2R2).

Hardware Support

Support of new PCIe functions:

Two new PCIe function types are supported - ISM and RCE.

1. An ISM (Internal Shared Memory) network adapter allows a virtual PCIe function requiring a unique VCHID with up to 255 virtual function Ids. The ISM network adapter allows one PNETID to be specified which has to be identical for all virtual functions of a VCHID.

2. An RCE (Regional Crypto Enablement) function is used for all IBM approved vendor crypto adapters.

PCIe enhancement:

The definition of PCIe functions for virtual functions of an adapter is now easier because 'Add PCIe function' offers a new field 'Number of virtual function' which helps to define multiple functions for one CHID at a time by incrementing the Virtual Function ID.

Validation enhancement:

At Build production time it is checked if for those PCIe functions using PNETIDs a channel path exists which is accessing at least one identical LPAR and has the same PNETID defined. Chpids and PCIe functions requiring a VCHID (currently CHPIDs of type IQD and functions of type ISM) now require a processor unique PNETID.

Processor support:

HCD supports the IBM z13 processor family:

- processor types 2964-N30, -N63, -N96, -NC6, -NE1 (new support level H160310)
- processor types 2964-L30, -L63, -L96, -LC6, -LE1 (support level H160310)
- processor types 2965-N10 and N20 (support level H160310)
- processor types 2965-L10 and L20 (support level H160310).

Summary of changes for z/OS Version 2 Release 2 (V2R2)

The following changes are made for z/OS® Version 2 Release 2 (V2R2).

z/OS V2R2 is planned to be the last release to include the LDAP back-end support for HCD. Users will no longer be able to use the LDAP interface to modify the content of an HCD IODF.

For a description of the new and changed support for V2R2, see “Summary of changes for z/OS Version 2 Release 1 (V2R1) as updated February, 2015.”

Summary of changes for z/OS Version 2 Release 1 (V2R1) as updated February, 2015

New Hardware Support

Support 6 Channel Subsystems and 4 Subchannel Sets

For the new processor type the maximum number of channel subsystems is extended to 6 and the number of subchannel sets is increased. See “Defining channel subsystems” on page 93 for more information.

PCIe function enhancements

With processor IBM z13™ the PCIe functions of type ROCE now require a virtual number definition and allow only 2 PNET ID for its external physical network assignments. See “Working with PCIe functions” on page 103 and “Defining PCIe functions” on page 103 for more information.

VCHID support for internal channels

For internal channel paths, like IQD, there does not exist a physical correspondence to hardware, hence there does not exist a PCHID value.

Instead a unique virtual channel ID (VCHID) is assigned. A VCHID value can arbitrarily be selected out of the supported range for valid PCHID values. See “PCIe function” on page 304 for more information.

New CS5 chpid type

A new CHPID type CS5 based on PCIe technology is supported. See “Defining or editing channels using Host Communication Adapters:” on page 115 for more information.

Processor support

HCD supports the IBM IBM z13 processor family (processor types 2964-N30, -N63, -N96, -NC6, -NE1).

Summary of changes for z/OS Version 2 Release 1 (V2R1)

Enhancements of the I/O Autoconfiguration function

HCD provides the following enhancements of the I/O Autoconfiguration function that has been introduced in z/OS V1R12:

- In addition to switched FICON connected controllers, I/O Autoconfiguration can now discover FICON directly attached controllers and devices and proposes point-to-point connection paths if available.
- I/O Autoconfiguration supports the inclusion or exclusion of specific switches or CHPIDs into the discovery and proposal process, that users can explicitly specify with the invocation of an I/O Autoconfiguration request. For this purpose, HCD introduces four new autoconfiguration policy keywords:
 - AUTO_CHPID_INCLUDE
 - AUTO_CHPID_EXCLUDE
 - AUTO_SWAD_INCLUDE
 - AUTO_SWAD_EXCLUDE
- The autoconfiguration policy keyword AUTO_SS_DEVNUM_SCHEME accepts a new value NONE. This value bypasses control unit and device number proposals by HCD and lets the user manually apply the numbers for detected objects.
- I/O Autoconfiguration allows discovery by controller serial number and filters the discovered controllers accordingly.
- HCD can process an I/O Autoconfiguration request that is partially directed against unavailable systems of an LPAR group or a sysplex, or against systems that are not capable to support I/O Autoconfiguration. Users can specify that the request applies to appropriate systems only, and that unavailable/uncapable systems are tolerated but ignored.
- HCD allows users to change certain I/O Autoconfiguration policies between two subsequent controller discoveries without the need to make a new fabric discovery. This enables I/O Autoconfiguration to perform each new controller discovery with changed policies.
- HCD provides the SAVE command on the I/O Autoconfiguration list panels (*Discovered New or Changed Controller List*, *Proposed Control Unit List*, and *Proposed Control Unit / Device List*) to allow users to save the fabric and controller discovery results and proposals in a data set.

HMC-wide activate

You can use a new HMC-wide activate function of HCD to remotely distribute and activate a new production IODF from a single managing z/OS system on all target systems of those CPCs that are configured in the Hardware Management Console (HMC) and that are defined in a specified TCP/IP connection table.

Launch this function from the z *Systems Cluster List* using action *Work with CPC images* introduced in z/OS V1R13. The upcoming *CPC Image List* now displays the activation status of the connected z/OS and z/VM systems and provides new actions to activate the accessed production IODF for hardware and/or software changes at the selected z/OS or z/VM system. The new production IODF is sent to the target system if necessary. You can remotely issue any operational commands that are required for the activation. The messages resulting from the activation or from processing system commands are displayed in a message list.

For the HMC-wide activate function on remote z/OS and z/VM systems, HCD to HCD communication needs to be set-up. For this communication, the HCD agent is used. In previous releases, the HCD agent was only required when using HCM. The description of this set-up is now provided as a copy and adaption from the *z/OS and z/VM HCM User's Guide* in Appendix E, "Establishing the host communication," on page 505.

Support of PCIe functions

Peripheral Component Interconnect Express (PCIe) adapters offer new functionality to systems running on IBM zEnterprise EC12 and BC12 (zEC12 and zBC12) processors in order to connect, for example, to an IBM zEnterprise BladeCenter Extension (zBX). Therefore, HCD introduces a new dialog where users can define PCIe functions, assign them to LPARs, and activate them via IOCP or dynamically.

In addition, HCD provides the following new reports:

- The *PCIe Function Summary Report* displays the partitions in the access and candidate lists which are entitled to access the available PCIe functions.
- The *PCIe Function Compare Report* shows the changes of PCIe functions between processors of two IODFs.

In addition, HCD supports the new I/O configuration statement FUNCTION for defining and configuring PCIe functions.

Validation enhancements

HCD implements several new validation checks to help users to avoid unintended results:

New warning message when a CF coupling connection changes due to connectivity updates

HCD issues a new warning message CBDG422I, when users add or delete a CSS to/from a coupling CHPID that involves a change of the coupling facility connection, affecting the definition of the connected processor. This message informs the user about the change and a potentially required dynamic activation of the target processor.

Warning message CBDA845I now also issued for ACTIVATE SOFT system command

For users of the HCD *Activate ...* dialogs, if required, HCD issues warning message CBDA845I when users specify an ACTIVATE software-only request without hardware validation, because this will not process involved changes to coupling facility control units and devices for the software. This message is now also issued, if users specify an ACTIVATE SOFT system command without hardware validation.

Enhanced CF Channel Path Connectivity List

A new column in the *CF Channel Path Connectivity List* shows for both the source and destination channel path either the physical channel identifier (PCHID) to which the channel path is assigned or its host communication adapter ID and port number.

OS group change action available for device groups

The *OS group change* action up to now has been available from the *I/O Device List* showing single devices only. Starting with this release, this action is also available from the *I/O Device List* showing device groups.

HCD batch enhancements

HCD provides the following new batch utility features:

Filter parameters for graphical reports created via batch utility

As with the HCD dialog for creating graphical reports, you now can specify filter parameters when creating graphical reports with the batch utility function.

ACTIVATE command now available as an HCD batch command

Users can now issue the ACTIVATE command as an HCD batch command. The syntax is the same as described in *z/OS MVS System Commands*.

New profile options

There are the following new keywords that you can specify in the HCD profile for the following purposes:

- **Unconditional generation of D/R site OS configurations:** Use profile option UNCOND_GENERATE_DROS to regenerate D/R site OS configurations whenever a new production IODF is built, independent from whether the configurations have been previously modified or not.
- **Specify remote call connection table:** Use profile option CONNECTION_TABLE to specify the name of a data set that contains the connection table for establishing connectivity to the remote systems while working with CPC images.
- **Enable remote call logging:** Use profile option RCALL_LOG to activate logging of remote calls into a data set while working with CPC images.
- **Set initial remote call timeout value:** Use profile option RCALL_TIMEOUT to set the timeout value for the initial connection to a remote system when working with CPC images.

CHID Summary Report

The *CHID Summary Report* as part of the *CSS Summary Report* lists all defined channel paths and PCIe functions grouped by their defined PCHID values or, as applicable, by their HCA adapter or port IDs.

Verify a configuration by means of I/O Autoconfiguration (zDAC)

With HCD you can now verify the active or target configuration by means of z/OS discovery and I/O Autoconfiguration (zDAC), if Tivoli System Automation (TSA) I/O operations is not installed or not working. This is possible for a processor supporting I/O Autoconfiguration and for a system in the local sysplex, which is capable for dynamic activates. The verification is limited to FICON attached storage devices.

When generating the I/O path report, HCD includes information about single point of failures (SPOFs) into the sensed data if the report is produced for the local system. This is done when getting the report via (TSA) I/O operations as well as via zDAC.

Hardware support

HCD supports the IBM zEnterprise EC12 and BC12 (zEC12 and zBC12) processor family (processor types 2827-H20, -H43, -H66, -H89, -HA1 and 2828-H06, -H13).

Chapter 1. Hardware configuration definition - What is it?

Overview

This topic explains:

- What HCD is and how it differs from MVSCP and IOCP
- What HCD offers you
- How HCD works
- The environment in which HCD operates

What HCD is and how it differs from MVSCP and IOCP

The channel subsystem (CSS) and the IBM z/OS operating system need to know what hardware resources are available in the computer system and how these resources are connected. This information is called *hardware configuration*.

Hardware Configuration Definition (HCD) provides an interactive interface that allows you to define the hardware configuration for both a processor's channel subsystems and the operating system running on the processor.

Before HCD was available, you had to use IOCP to define the hardware to the channel subsystem and the MVS Configuration Program (MVSCP) to define the hardware to the MVS operating system. The following sections explain in what way HCD differs from MVSCP and IOCP when defining, validating and reconfiguring configuration data.

Definition of configuration data

This topic informs about the differences between MVSCP and IOCP on the one hand and HCD on the other hand when performing the task of defining configuration data.

How MVSCP and IOCP worked

With MVSCP and IOCP you were limited to defining one processor or operating system per input data set. This meant that you needed more than one data set when you used MVSCP or IOCP.

Figure 1 on page 2 illustrates the definition process using several sources for writing and modifying the hardware configuration using IOCP and MVSCP data sets.

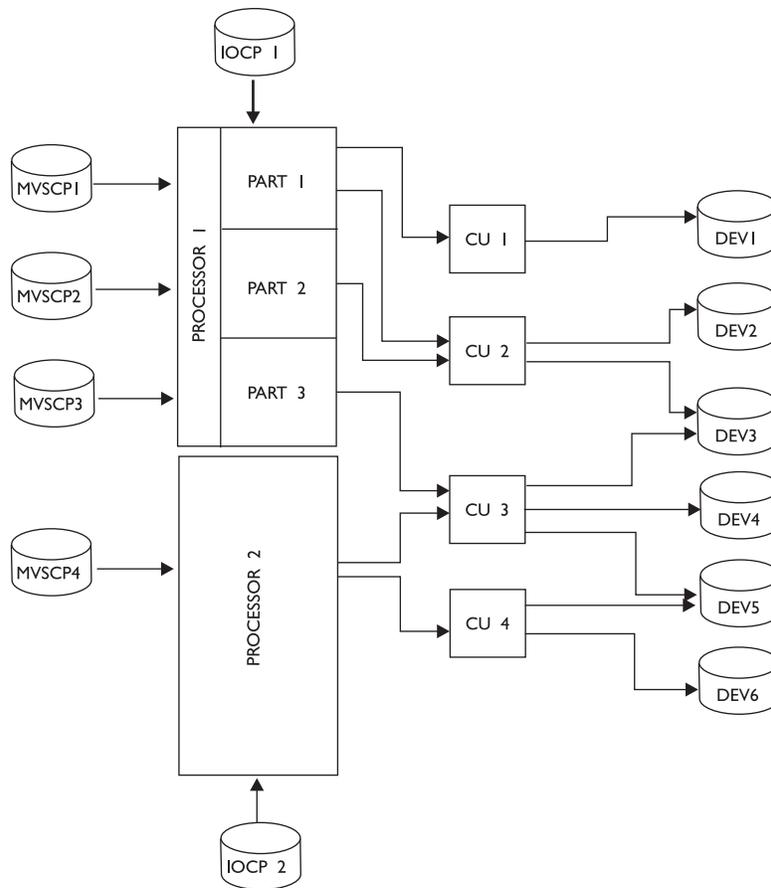


Figure 1. Multiple data set configuration definition without HCD

What HCD does

The configuration you define with HCD may 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 (IODE)*. The IODE as single source for all hardware and software definitions for a multi-processor system eliminates the need to maintain several independent MVSCP or IOCP data sets. That means that you enter the information only once using an interactive dialog.

Figure 2 on page 3 illustrates the definition process using one source for writing and modifying configuration data in the IODE.

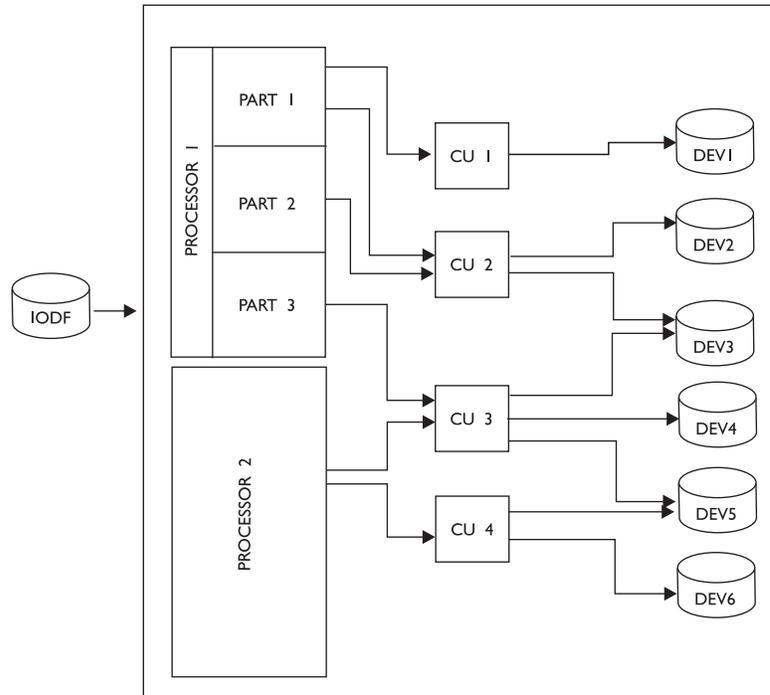


Figure 2. Single data set configuration definition with HCD

Validation of configuration data

How MVSCP and IOCP worked

MVSCP and IOCP were separately running independent programs. Prior to IPL it was not checked whether the MVSCP output matched the configuration in the I/O configuration data set (IOCDS). Even if the definitions of both programs were not identical, it was possible for an IPL to be successful if the devices needed to start the system were included in both programs. Therefore, discrepancies would be detected only after the system had been running for some time. Such a discovery could have happened at a very inconvenient moment.

What HCD does

The data entered with HCD is validated and checked for consistency and completeness. Because the check is performed when the data is defined, rather than when the device is accessed, inconsistencies can be corrected right away, and unplanned system outages resulting from inconsistent definitions can be avoided.

If you include ESCON or FICON Director definitions in the IODF, HCD also validates the switch port usage and connection information for all devices and channels connected to each of the directors in the configuration. In addition, HCD validates the complete path from the processor through the switch to the control unit and device.

Planned I/O paths, as defined in an IODF, can be checked against the active configuration on the system. Discrepancies are indicated and can be evaluated before and after the IODF is activated.

Reconfiguration of configuration data

How MVSCP and IOCP worked

The IOCP updated the input/output configuration data set (IOCDS) that resided in the hardware support processor. This information was loaded into the hardware system area during power-on reset (POR). If the configuration was changed, it was necessary to write a new IOCDS using IOCP and to load it into the hardware system area with a POR.

MVSCP created the control information (such as UCBs, EDTs, and NIPCONs) needed by MVS to describe the hardware configuration and stored this information in the SYS1.NUCLEUS data set. The nucleus information was loaded at IPL time into storage. If a change was made to the I/O configuration, it was necessary to IPL to make the information available to MVS.

What HCD does

Dynamic reconfiguration management is the ability to select a new I/O configuration during normal processing and without the need to perform a power on reset (POR) of the hardware or an initial program load (IPL) of the z/OS operating system.

The ability of HCD to provide equivalent hardware and software I/O definitions and to detect when they are not in sync is essential for dynamic I/O reconfiguration management. HCD compares both the old and the new configuration and informs the hardware and software about the differences. You may add, delete, and modify definitions for channel paths, control units and I/O devices without having to perform a POR or an IPL.

What HCD offers you

This section summarizes what you can do with HCD and how you can work with HCD.

Single Point of Control: With HCD you have a single source, the IODF, for your configuration data. This means that hardware and software definitions as well as ESCON or FICON director definitions can be done from HCD and can be activated with the data stored in the IODF.

Increased System Availability: HCD checks the configuration data when it is entered and therefore reduces the chance of unplanned system outages due to inconsistent definitions.

Changing Hardware Definitions Dynamically: HCD offers dynamic I/O reconfiguration management. This function allows you to change your hardware and software definitions on the fly - you can add devices, or change devices, channel paths, and control units, without performing a POR or an IPL. You may also perform software-only changes, even if the hardware is not installed.

Sysplex Wide Activate: HCD offers you a single point of control for systems in a sysplex. You can dynamically activate the hardware and software configuration changes for systems defined in a sysplex.

Migration Support: HCD offers a migration function that allows you to migrate your current configuration data from IOCP, MVSCP, and HCPRIO data sets into HCD. Migration support also allows you to make bulk changes to the configuration using an editor on the IOCP/MVSCP/HCPRIO macro statements.

Accurate Configuration Documentation: The actual configuration definitions for one or more processors in the IODF are the basis for the reports you can produce with HCD. This means that the reports are accurate and reflect the up-to-date definition of your configuration.

HCD provides a number of textual reports and graphical reports, that can be either printed or displayed. The printed output can be used for documentation purposes providing the base for further configuration planning tasks. The display function allows you to get a quick overview of your logical hardware configuration.

Guidance through Interactive Interface: HCD provides an interactive user interface, based on ISPF, that supports both the hardware and the software configuration definition functions. The primary way of defining the configuration is through the ISPF dialog. HCD consists of a series of panels that guide you through all aspects of the configuration task. The configuration data is presented in lists.

HCD offers extensive online help and prompting facilities. Help includes information about panels, commands, data displayed, available actions, and context-sensitive help for input fields. A fast path for experienced users is also supported.

Batch Utilities: In addition to the interactive interface, HCD also offers a number of batch utilities. You can use these utilities, for instance, to migrate your existing configuration data; to maintain the IODF; or to print configuration reports. For a complete list of batch utility functions, refer to Chapter 13, "How to invoke HCD batch utility functions," on page 327.

Cross Operating System Support: HCD allows you to define both MVS type (for example OS/390 or z/OS) and VM type configurations from z/OS and to exchange IODFs between z/OS HCD and z/VM HCD.

Support of z Systems clusters: HCD provides single point of control functions in a z Systems cluster for dynamic I/O configuration changes and for the management of IOCDs and IPL attributes. These functions assist users in configuring and operating those processors that are configured in a z Systems cluster controlled by the same Hardware Management Console (HMC).

LDAP Interface Capability: HCD provides you with search and update capabilities for IODF data via an LDAP interface.

How HCD works

HCD stores the hardware configuration data you defined in the IODF. A single IODF can contain definitions for several processors (or LPARs) and several MVS or VM operating systems. It contains all information used to create IOCDs and the information necessary to build the UCBs and EDTs. When HCD initiates the function to build the IOCDs, the IODF is used as input. The IOCDs with the channel subsystem definitions of a processor is then used to perform POR. The same IODF is used by MVS to read the configuration information directly from the IODF during IPL. If your environment includes z/OS and z/VM on different processors or as logical partitions on the same processor, the IODF can also be used to document the z/VM configuration.

Figure 3 on page 6 shows an example of a configuration using HCD.

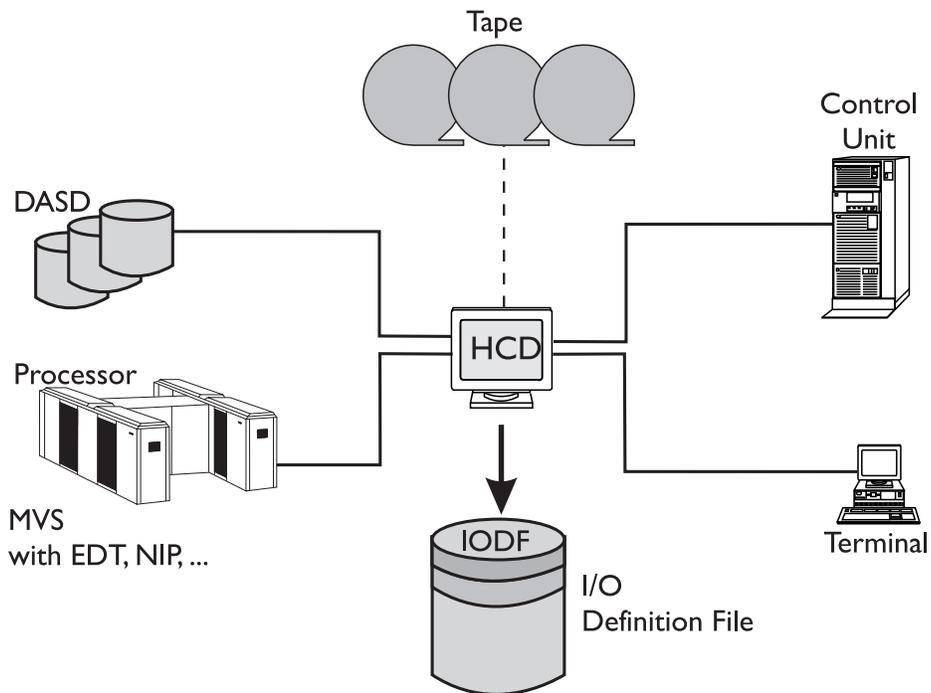


Figure 3. Configuration definition with HCD

Objects managed in the IODF

HCD lets you define the configurations as objects and their connections. The following objects and their connections are managed by HCD:

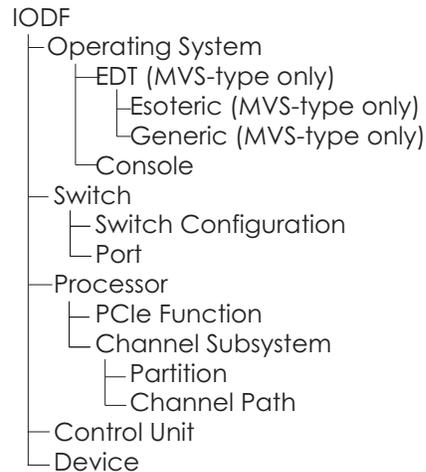


Figure 4. Objects managed by HCD

For all these objects the HCD dialog provides action lists where you can define the characteristics and the relation between the objects.

IODF used at IPL

After you complete the input of your configuration data, you have to build a production IODF. The production IODF is used by the operating system to build

the configuration data (for example, control blocks) at IPL time. This active production IODF is also used for building the IOCDs.

Figure 5 illustrates the build phase of a production IODF and of an IOCDs.

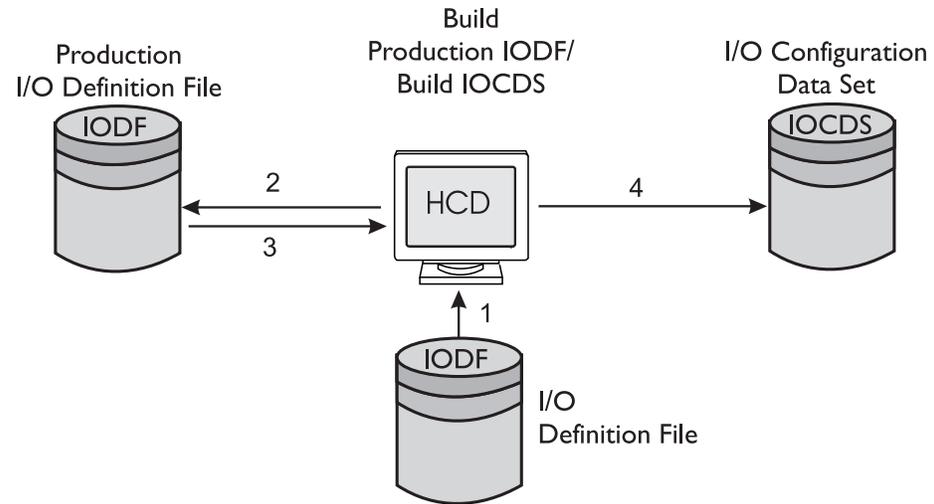


Figure 5. Building a production IODF and an IOCDs

The production IODF cannot be updated (read-only). This ensures that the data in the production IODF used at IPL remains the same during the run time of that system.

Relationship of data sets used by HCD

Figure 6 on page 8 shows the relationship between the data sets used by HCD and how you can work with or change these data sets.

By using the define and modify tasks or by migrating MVSCP, IOCP, and HCPRIO input data sets you create a work IODF. After finishing the definition, you build a production IODF from your work IODF, which you can use to IPL your system or to activate your configuration dynamically. The data sets shown at the bottom of the figure are created by the appropriate tasks like Build IOCDs, Build IOCP input data set, Build HCPRIO input data set, and Activate switch configuration.

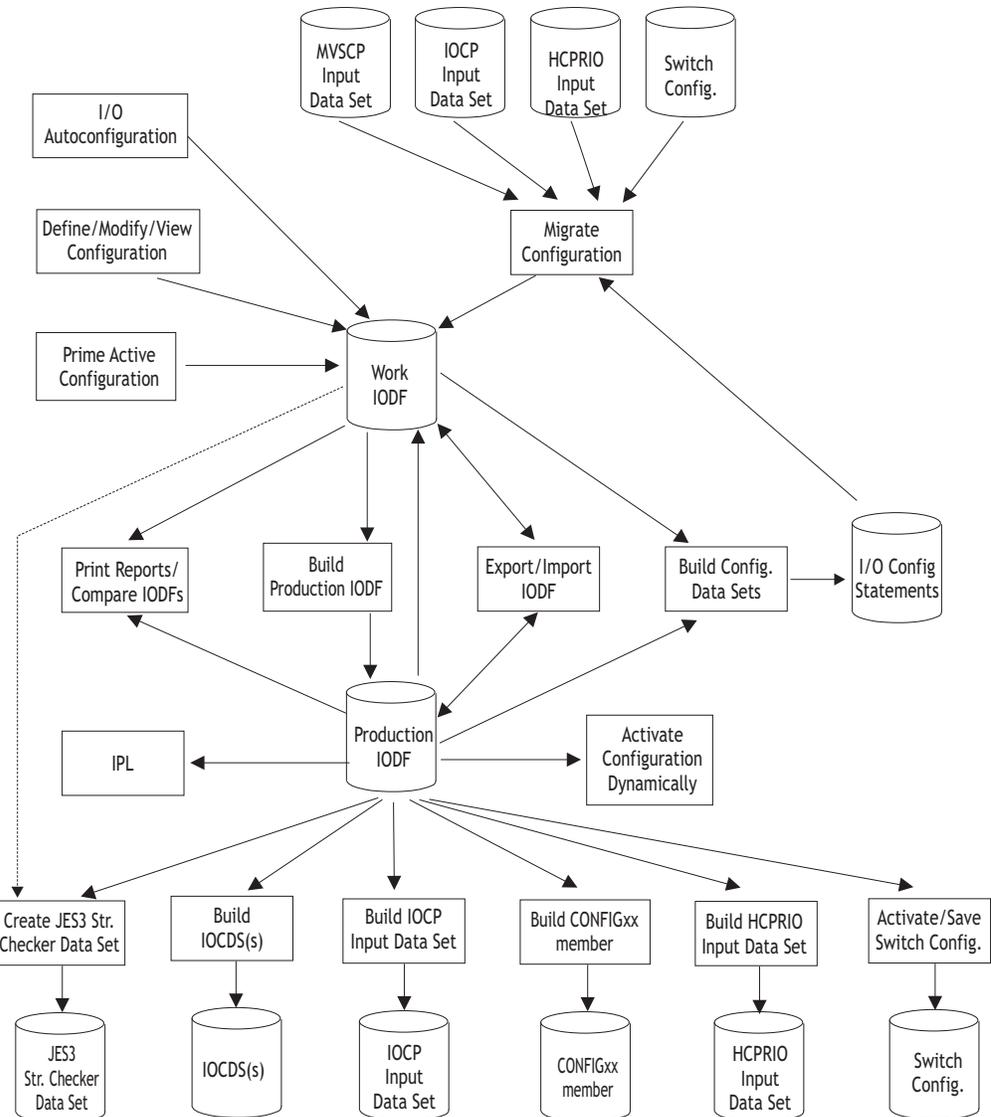


Figure 6. Relationship of data sets used by HCD

The environment in which HCD operates

HCD is part of z/OS. It needs a running z/OS system before it can be used to define a hardware configuration. Therefore, an installation should first load a z/OS system, using an old IODF, or a ServerPac Starter IODF to IPL the z/OS system for the first time. HCD can then be used on that system to define the full configuration.

HCD uses the unit information modules (UIMs) of z/OS. UIMs contain device dependent information, such as parameters and features of devices. The UIMs must be installed in the z/OS system before you use HCD to define a configuration. The UIMs are also used at IPL time to build the UCBs. That is why they have to be installed in SYS1.NUCLEUS at IPL time.

UIMs are provided for the IBM devices supported by z/OS, OS/390, or MVS by the device product owners. You can write your own UIMs for non-IBM devices. To get information about UIMs, see *z/OS MVS Device Validation Support*.

HCD and I/O Operations

The ESCON and FICON architectures are supported by a class of devices called Directors (switches) that manage the switching functions. These switches may be connected to a processor (via channel path), a control unit, or another switch. The switches connect channel paths and control units only for the duration of an I/O operation. The “internal” configuration of a switch is called switch configuration.

The I/O Operations component of Tivoli System Automation for z/OS (TSA for z/OS, formerly known as ESCON Manager) maintains and manages switch configurations.

I/O Operations functions, such as the activation of switch configurations and the retrieval of the active configuration data, can be invoked from HCD. This gives you a single point of control for all switch configuration activities, as well as the possibility to check whether a certain data path (from processor to device) is fully configured or not.

Moreover, you can use HCD to migrate switch configurations into HCD from three different sources: directly from the switches, from a saved switch file, or from ISPF tables saved by I/O Operations. You also can save switch configuration data in a switch file, as well as activate the switch configuration. Thus, the switch is activated using the switch configuration stored in the IODF.

I/O Operations permits retrieval of information from the active system. Parts of this information (serial numbers, VOLSERs, port names) can be provided by HCD via the functions *Priming*, *Sensing*, *Activate Verification*, and *Prompting for VOLSER*. I/O Operations must be installed for this purpose, even if there are no switches in the configuration.

Prerequisites

For information on general installation requirements, refer to the PSP Bucket for the latest information about the prerequisites. Also consider the following prerequisites and recommendations:

- For migrating existing switch configurations and activating switch configurations, Tivoli System Automation for z/OS (I/O Operations), must be installed and active.
- If TSA for z/OS (I/O Operations) is not installed for verifying the I/O configuration or generating the *I/O Path Report* for a system of the local sysplex, a subset of the function is available for FICON attached storage devices by means of the zDAC function. In order to use zDAC for this purpose, discovery data must be available from the LPARs that you wish to verify.

If you want to prime, then in order to obtain the data for a system in a sysplex, there must also be a VTAM session between the local system and the target system. The target system must have I/O Operations installed and running.

HCD and the coupling facility

HCD provides the user interface to support processors that have coupling facility capability.

The coupling facility itself is implemented as an extension to PR/SM features on selected processors (refer to *z/OS HCD Planning* for a list) and runs in a PR/SM partition. It enables direct communication between processors through a specific communication partition (coupling facility partition), connected by coupling facility channels.

You use HCD to specify whether a logical partition is running a coupling facility or an operating system. New channel path definitions in the IODF are used to connect a coupling facility-capable processor to a coupling facility partition:

- The coupling facility receiver channel (CFR channel) path that accesses the partition the coupling facility is running on.
- The coupling facility sender channel (CFS channel) path that accesses the partition the operating system is running on.
- The coupling facility peer channel path that accesses either partition bidirectionally on IBM zSeries processors or their successors.

HCD automatically generates the coupling facility (CF) control unit and devices that are necessary for IOCP processing. Figure 7 shows a processor configuration with coupling facility implemented.

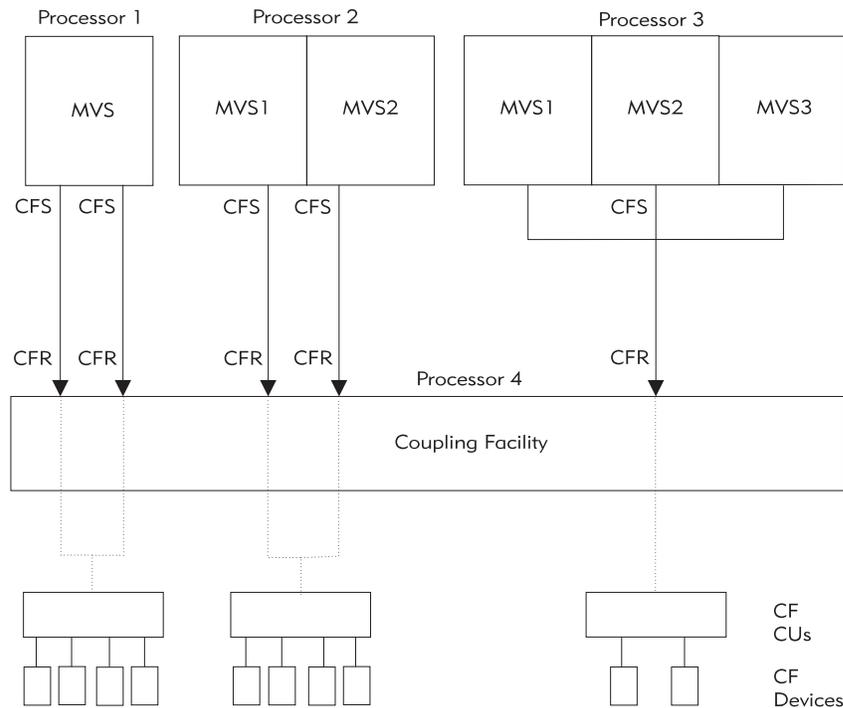


Figure 7. Processor configuration with coupling facility implemented

HCD enables you to dynamically reconfigure the coupling facility channels that are connected to the operating system partition.

Note: With CF duplexing, a CF logical partition can use the coupling facility sender function to communicate with another CF logical partition. That means, you can define sender channel paths (CFS, CBS, ICS) besides the receiver channel paths (CFR, CBR, ICR) in a CF partition.

For more information on HCD and coupling facility, refer to *z/OS Parallel Sysplex Overview*.

Support of z Systems clusters

Note on terminology:

The term z Systems cluster, used in the HCD dialog and throughout this documentation, refers to central processor complexes (CPCs) controlled through the Hardware Management Console (HMC).

HCD allows you to define and control configuration data for each CPC that is configured in a z Systems cluster. You use HCD for those CPCs that can have their IOCDS and IPL attribute management and their dynamic I/O configuration changes controlled remotely:

- Writing IOCDSs
- Managing write-protection
- Marking the IOCDS as active POR IOCDS
- Updating IPL address and LOAD parameter values
- Distributing and activating new production IODFs on connected target systems

HCD displays all CPCs that are configured in a z Systems cluster and controlled by the Hardware Management Console (HMC). The CPCs and the HMC must be connected to the same management network. A CPC is identified by the system network architecture (SNA) address of its support element, which is specified when the processor complex is configured on the HMC. HCD uses the SNA address to be able to write and manage IOCDSs from any processor in a z Systems cluster, and to view and update IPL attributes.

From the **z Systems Cluster List** on a single point of control, users can select an option to list all logical partitions (images) belonging to the current CPC along with various information for each partition, for example, their operation status (deactivated, activated, IPLed), the system name, sysplex name, operating system type and release level.

If users provide a connection table that contains the IP addresses and user logon data on the target systems, HCD shows the corresponding activation status and allows distributing and activating new production IODFs on those systems.

Support of the sysplex environment

If you have interconnected systems, it is important to have a single point of control for systems in such a sysplex environment. HCD offers the support of the sysplex environment in several ways:

- You can define all processor and operating system configurations in one IODF.
- After a complete definition you can download the IOCDSs for all processors. This can be initiated from the controlling HCD.
- If you want to change configurations dynamically within the sysplex, you can initiate the activation of the hardware and software configuration changes for systems defined in a sysplex from the controlling HCD.

In addition, HCD offers a function that compares the active I/O configuration against the defined configuration in an IODF. This function can produce reports or lists of I/O paths to show the sensed data against the logical definitions of the paths in the IODF. HCD provides this function to get the data for your local system and for systems in a sysplex.

Based on a production IODF, HCD can also build CONFIGxx members for your local system or for systems in the sysplex.

For systems which are members of a sysplex, CONFIGxx members can be verified against selected systems. Responses are displayed in a message list.

Refer to Chapter 13, “How to invoke HCD batch utility functions,” on page 327, if you want to run batch jobs in a sysplex environment.

Chapter 2. Migration

Overview

This topic explains:

- An overview of the migration process
- Steps needed for migrating to HCD for z/OS Version 2 from an earlier HCD release

Migration overview

Your plan for migrating to the new level of HCD should include information from a variety of sources. These sources of information describe topics such as coexistence, service, hardware and software requirements, installation and migration procedures, and interface changes.

The following documentation provides information about installing your z/OS system. In addition to specific information about HCD, this documentation contains information about all of the z/OS elements.

- *z/OS Planning for Installation*

This book describes the installation requirements for z/OS at a system and element level. It includes hardware, software, and service requirements for both the driving and target systems. It also describes any coexistence considerations and actions.

- *ServerPac Installing Your Order*

This is the order-customized installation book for using the ServerPac Installation method. Be sure to review “Appendix A. Product Information”, which describes data sets supplied, jobs or procedures that have been completed for you, and product status. IBM may have run jobs or made updates to PARMLIB or other system control data sets. These updates could affect your migration.

- *z/OS Migration*

This document describes how to migrate to the current z/OS version and release from previous releases for all z/OS elements and features and also includes migration actions for HCD and HCM.

“Migration roadmap”

- identifies the migration paths that are supported with the current level of HCD
- describes the additional publications that can assist you with your migration to the current level

Migration roadmap

This section describes the migration paths that are supported by the current release of HCD. It also provides information about how to migrate to the current HCD release from previous releases.

You can find further migration information in *z/OS Migration*.

Security migration

With z/OS V2R1, HCD uses the application ID CBDSERVE to verify any user that logs on to the HCD agent, this is, any user that uses HCM to perform hardware configuration definitions. If you have the APPL class active in your external security manager, for example in RACF, and you have a generic profile in that class that covers the new HCD application ID CBDSERVE, you need to permit all HCM users READ access to that profile. Otherwise, the users of HCM are no longer able to log on to HCD. If you use the existing generic APPL profile for other purposes, you may define your own CBDSERVE profile in the APPL class to control access specific for HCD. For details see “Controlling access to HCD services” on page 507.

Migration tasks

The following sections contain additional migration procedures or information:

- “Upgrade an IODF” on page 44
- “IODF release level compatibility” on page 53
- “Migrating existing switch configurations” on page 181
- “IOCP input data sets using extended migration” on page 221
- Chapter 12, “How to migrate existing input data sets,” on page 279
- “Upgrade IODF” on page 330
- “Migrate I/O configuration statements” on page 331

Information on IODF coexistence can be found in “IODF release level compatibility” on page 53.

Chapter 3. How to set up, customize and start HCD

Overview

This information unit handles the following topics:

- “Setting up HCD”
- “Tailoring the CLIST CBDCHCD” on page 17
- “Starting and ending HCD” on page 18
- “Defining an HCD profile” on page 19
- “Customizing HCD EXEC procedures” on page 30

Setting up HCD

HCD is a base element of z/OS and therefore installed with the z/OS product. For more information, refer to *z/OS Planning for Installation*.

The installation of HCD is carried out using SMP/E. The install logic and the JCLIN are provided by HCD.

Setting up HCD requires the following steps:

1. Install z/OS with the HCD FMIDs.
2. Install other products that are required for HCD (refer to *z/OS Planning for Installation*).
3. Before you start HCD, you have to set up the load libraries that contain the HCD help modules. You can achieve this in one of the following ways:
 - Include SYS1.SCBDHENU (or SYS1.SCBDHJPN for Kanji) in the linklist concatenation (LNKLSTxx member), or
 - Allocate data set SYS1.SCBDHENU (or SYS1.SCBDHJPN for Kanji) to ISPLLIB.

If you choose to access the libraries through the ISPLLIB concatenation, ISPLLIB must be allocated prior to invoking ISPF with the TSO ALLOC command or through a CLIST. ISPLLIB is used as a tasklib by ISPF as it is searched first.

When using the View graphical configuration report:

- Include the GDDM load library in the linklist concatenation (LNKLSTxx member).
- Allocate the GDDM sample data set:

For GDDM 2.1 and 2.2:

```
ALLOcAtE F(ADMPC) DSN('pplib.GDDM.GDDMSAM') SHR REUSE
```

For GDDM 2.3 or later:

```
ALLOcAtE F(ADMPC) DSN('pplib.GDDM.SGDDMSAM') SHR REUSE
```

If you are using a programmable workstation and you communicate with the host using a 3270 emulator session, the GDDM-OS/2 link files must be installed on your workstation. Note that the high-level qualifier for the GDDM data set might vary from installation to installation.

4. Allocate the data set SYS1.SCBDCLST to the SYSPROC ddname concatenation.

Note: SYS1.SCBDCLST has a fixed record format (RECFM=FB). If your other SYSPROC data sets have a variable record format (RECFM=V or VB), copy SYS1.SCBDCLST to a data set with variable record format. You have to remove sequence numbers (in the CLIST) after copying the members to a data set with variable record format.

5. For processing large IODFs, and when HCD option IODF_DATA_SPACE is set to N0, the size of your TSO region may not be sufficient. When you specify the region size on the TSO logon panel, calculate as follows:

$$2 \times \text{IODF size} + 4 \text{ MB}$$

Example:

Assumed IODF size: 8000 blocks, 4 KB per block = 32 MB

Suggested region size: 68 MB

6. For setup requirements when using HCD via the HCD LDAP backend, see Chapter 15, "How to provide LDAP support for HCD," on page 369.

To run HCD, the modules in SYS1.SCBDHENU (containing HCD help members) and SYS1.NUCLEUS (containing the UIMs) must be accessible. For the HCD dialog, you can achieve this in three ways:

1. Include SYS1.SCBDHENU in the linklist concatenation (LNKLSTxx member).
2. Include SYS1.SCBDHENU in the JOBLIB/STEPLIB concatenation of the TSO logon procedure.
3. Include SYS1.SCBDHENU in the ISPLLIB load library concatenation. If you include SYS1.SCBDHENU into the ISPLLIB concatenation, ISPLLIB must be allocated prior to invoking ISPF (in TSO or through JCL in the logon procedure). ISPLLIB is used as a tasklib by ISPF and is searched first. A pure LIBDEF for ISPLLIB does not suffice to invoke HCD.

HCD allocates SYS1.NUCLEUS automatically at initialization time if the keyword UIM_LIBNAME is not specified in the HCD profile. You may use the HCD profile to specify a different name and the volume serial number of the library that contains the UIMs (see also "Defining an HCD profile" on page 19). If you do not specify a name in the profile, SYS1.NUCLEUS is assumed as default name for the UIMs. For IPL, however, the UIMs and UDTs must be in SYS1.NUCLEUS.

The following HCD libraries are defined via the ISPF 'ISPEXEC LIBDEF' command if HCD is invoked via CLISTs CBDCHCD and CBDCHCD1:

- SYS1.SCBDPENU for panels
- SYS1.SCBDMENU for messages
- SYS1.SCBDTENU for tables

How to invoke HCD in dialog mode

To invoke HCD in dialog mode, ISPF must be active. After you have invoked ISPF, you can use the CLIST CBDCHCD to activate the HCD function. You may add HCD to an ISPF selection menu, for example, the ISPF/PDF Master Application Menu (ISP@MSTR), and invoke HCD using the CLIST CBDCHCD. Figure 8 on page 17 shows a sample panel that illustrates how to include HCD on the main ISPF/PDF panel. Alternatively, the CLIST can be invoked from the ISPF option 6, or from the command line.

Note that HCD must be invoked with the "NEWAPPL(CBD)" parameter in the CLIST CBDCHCD.

To ensure that the CBDCHCD CLIST can successfully allocate the following libraries, make sure that these libraries are cataloged:

- SYS1.SCBDPENU --> HCD Panel Library
- SYS1.SCBDMENU --> HCD Message Library
- SYS1.SCBDTENU --> HCD Table Library

Note: The HCD Panel, Message, and Table libraries are allocated by the CBDCHCD CLIST using the LIBDEF function of ISPF. If other ISPF Dialogs are using the LIBDEF function of ISPF, and you do not want HCD to overlay their allocations, you can update your ISPF startup by adding the HCD data sets to the ISPF ISPPLIB, ISPMLIB, and ISPTLIB concatenations.

```

%----- ISPF MASTER APPLICATION MENU -----
%OPTION ==>_ZCMD                                +USERID - &ZUSER +
%                                                    +TIME   - &ZTIME
%                                                    +TERMINAL - &ZTERM
%                                                    +PF KEYS - &ZKEYS
%
% 1 +ISPF/PDF - ISPF/Program Development Facility
% 2 +SMP/E    - System Modification Program/Extended
% 3 +HCD      - Hardware Configuration Definition
% 4 +SDSF    - SYSTEM Display and Search Facility
% 5 +RACF    - Resource Access Control Facility
% X +EXIT    - Terminate ISPF using list/log defaults
%
+Enter%END+command to terminate ISPF.
%
)INIT
  .HELP = ISP00005 /* Help for this master menu */
  &ZPRIM = YES /* This is a primary option menu */
)PROC
  &ZSEL = TRANS( TRUNC (&ZCMD, '.')
                1, 'PANEL(ISR@PRIM) NEWAPPL(ISR)'
                2, 'PGM(GIMSTART) PARM(&ZCMD) NOCHECK NEWAPPL(GIM)'
                3, 'CMD(CBDCHCD) PASSLIB'
                4, 'PGM(DGTFMD01) PARM(&ZCMD) NEWAPPL(DGT) NOCHECK '
                5, 'PANEL(ICHPO0) NEWAPPL(RACF)'
                ' '
                ' '
                ' '
                ' '
                ' '
                X, 'EXIT'
                *, '?' )
  IF (&ZCMD = '%VSDSF')
    &ZSEL = 'PGM(ISFISP) NEWAPPL(ISF)'
)END

```

Figure 8. Sample ISPF Master Application Menu

Tailoring the CLIST CBDCHCD

A sample CLIST CBDCHCD (CBDCHCDJ for Kanji) is provided in SYS1.SCBDCLST to assist you in invoking HCD from the ISPF dialog. It allocates the HCD message log file (HCDMLOG), the trace data set (HCDTRACE), and the HCD term file (HCDTERM) with a default high-level qualifier of the userID (&SYSUID.). The CLIST also allocates the HCD data sets SYS1.SCBDPENU, SYS1.SCBDMENU, and SYS1.SCBDTENU (or SYS1.SCBDPJPN, SYS1.SCBDMJPN, and SYS1.SCBDTJPN for Kanji).

In CBDCHCD and CBDCHCDJ, the ISPEXEC LIBDEF statement for ISPPLIB, ISPTLIB and ISPMLIB is done with the STACK option. This leaves existing LIBDEFs untouched, so that after exit, the existing HCD libraries ISPPLIB, ISPTLIB and ISPMLIB are freed again.

If the prefixes for message, trace, and term data sets do not conform to the installation conventions, you may tailor the CLIST to match your installation

defaults. If you want HCD to use your TSO prefix as the high level qualifier, you can call CBDCHCD with the parameter NOPREF(YES). This causes HCD to use the qualifiers &PREFIX.&SYSUID.

CBDCHCD invokes another CLIST, CBDCHCD1. You may tailor this CLIST as well.

In any case, use the application ID for HCD: NEWAPPL(CBD).

The CLIST also tailors the ISPF environment by:

- Setting PFSHOW on. This forces all 24 function keys to be shown (if ISPF is defined to show 24 function keys).
- Setting lower PFKEYS as primary function keys.

Note: HCD can be invoked with the activated TRACE option, when you have specified parameter TRACE(YES) in the default CLIST. The size of the trace data set can be changed by modifying the CLIST. In addition, you can delete the HCD provided trace data set and allocate one according to your specific needs.

Starting and ending HCD

You start HCD like any other ISPF application in your TSO/E system. The procedure for starting an application is different for each installation but you can probably select HCD from a menu of applications that are available in your system. This causes a TSO/E CLIST to be executed. The sample CLIST that is supplied with HCD is CBDCHCD in library SYS1.SCBDCLST.

After you start HCD, the first panel that you see contains a menu of the HCD primary tasks:

```
z/OS V2.1 HCD
Command ==> _____

Hardware Configuration

Select one of the following.

1_ 0. Edit profile options and policies
   1. Define, modify, or view configuration data
   2. Activate or process configuration data
   3. Print or compare configuration data
   4. Create or view graphical configuration report
   5. Migrate configuration data
   6. Maintain I/O definition files
   7. Query supported hardware and installed UIMS
   8. Getting started with this dialog
   9. What's new in this release

For options 1 to 5, specify the name of the IODF to be used.

I/O definition file . . . 'SYS1.IODF00.HCD.WORK'          +

F1=Help   F2=Split   F3=Exit   F4=Prompt   F9=Swap   F12=Cancel
F22=Command
```

Figure 9. HCD **Primary Task Selection** panel

If you select task "1. Define, modify, or view configuration data", as shown in Figure 9 and press the Enter key, you trigger this task using the IODF 'SYS1.IODF00.HCD.WORK'.

To end an HCD session, either return to the *Primary Task Selection* panel and press the F12=Cancel key or the F3=Exit key twice or use the fast path command GOTO X.

Note:

1. Chapter 13, “How to invoke HCD batch utility functions,” on page 327 explains how you can invoke HCD from another program using the HCD programming interface.
2. To see a description of new functionality of the current release, select option 9. **What's new in this release.** Here you may find information about SPEs that are delivered after the completion of this document.

Defining an HCD profile

Before you start HCD, you can define an HCD profile to tailor HCD supplied defaults and processing options to your specific installation needs. Using a profile is optional and will remain in effect for the entire HCD session. At initialization time, HCD reads the profile and processes each statement in turn.

The profile statements are contained in a data set allocated to the DD name HCDPROF. The following statement allocates the profile data set to HCDPROF:

```
//HCDPROF DD DSN=&SYSUID..HCD.PROFILE,DISP=SHR
```

The data set must have the following characteristics:

- Be either a sequential data set or a member of a partitioned data set
- Have fixed-length, fixed-blocked record format
- Have 80 character records.

The HCD profile comprises the keywords described in “Keywords” on page 22 as well as policies for automatic I/O configuration described in Chapter 8, “How to work with I/O Autoconfiguration,” on page 187.

You can use the *HCD Profile Options* dialog (see “Working with the HCD Profile Options dialog” on page 20) to edit the profile keywords after having created and allocated the profile data set to HCDPROF.

In addition, you can define profile options manually in the profile data set.

You can extend a comment to the next line by using an asterisk (*) as a continuation character in column 1, as shown in the example:

```
MAP_CUTYPE=9000,NOCHECK /* map CU type 9000 to type NOCHECK */
```

or:

```
MAP_CUTYPE=9000,NOCHECK /* map CU type 9000 to type  
* NOCHECK */
```

However, be aware that the *HCD Profile Options* dialog truncates comments longer than 32 characters.

It is also possible to define profile options using inline statements in a batch job. The following example shows an inline profile definition:

```
//HCDPROF DD *
MIGRATE_EXTENDED = YES
VM_UIM = NO
/*
```

Note: When starting an HCD batch job from the dialog, the HCD profile data set is not passed automatically to the job but, if required, has to be specified in the JCL.

The following syntax rules apply to a profile statement:



keyword

is the name of the HCD keyword; each keyword starts on a new line.

= can be omitted, if the keyword is followed by a blank.

value

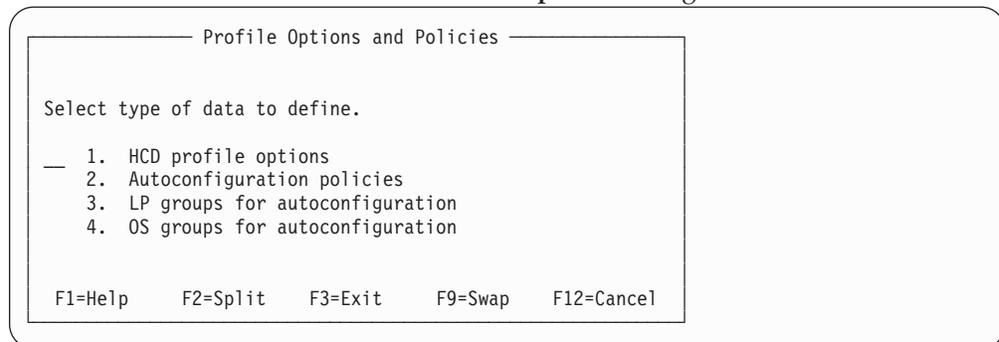
specifies one or more values to be assigned to the keyword.

Note:

1. A single statement must not exceed 72 characters.
2. Do not use sequence numbers in your HCD profile.
3. Use /* and */ as delimiters for comments in a profile statement.

Working with the HCD Profile Options dialog

You can maintain your HCD profile values in your profile data set using the HCD dialog. Select *Edit profile options and policies* from the HCD *Primary Task Selection* panel and then select **HCD profile options** from the *Profile Options and Policies* menu to invoke the **HCD Profile Options** dialog.



You can add, delete or modify keywords in your profile data set via this dialog. When you leave the HCD session, HCD writes all changes specified in this dialog to the profile data set.

In this dialog, all HCD profile keywords are listed in alphabetical order, followed to the right by their value and possibly a description (user comment). To see the *Description* column, scroll to the right (Shift + PF8).

- If a profile data set exists, HCD reads the contained keywords with their values and, if available, their descriptions. For keywords that are not explicitly defined in your profile data set, HCD shows the defaults.
- If you have no own profile data set allocated, this dialog lists the HCD default values. You cannot change these settings.

Note: The remainder of this section describes the use of the **HCD Profile Options** dialog with an allocated profile data set.

```

HCD Profile Options
Row 1 of 40 More: >
Command ==> _____ Scroll ==> CSR

Edit or revise profile option values.

HCD Profile : DOCU.HCD.PROFILE

/ Profile keyword      A Value +
# ACTLOG_VOL           Y *
_ ALLOC_SPACE          Y HCDASMP,60
_ ALLOC_SPACE          Y HCDRPT,60
# BATCH_IODF_NAME_CHECK Y NO
# BYPASS_UPD_IODF_FOR_SNA Y YES
# CHANGE_LOG           N NO
# CHECK_IODF           Y YES
# CHLOG_EXTENSION     Y 0
# CHLOG_VOL            Y *
# COLOR_BACKGROUND    Y _____
# COLOR_HIGH          Y RED
F1=Help      F2=Split      F3=Exit      F4=Prompt    F5=Reset
F7=Backward  F8=Forward    F9=Swap     F12=Cancel  F20=Right
F22=Command

```

If you change a displayed default value, HCD writes the changed entry into your data set.

The *HCD Profile Options* dialog accepts a description (comment) of maximum 32 characters. A comment must start on the same line as the keyword assignment, using the *’/*’* notation.

If you specified a longer comment in a manually edited profile data set, HCD shows the truncated comment only and also truncates the text in the profile data set the next time, HCD writes any modifications from the dialog.

For profile keywords which may occur multiple times with different value assignments, the dialog offers actions to add or delete selected entries (action codes *’a’* or *’d’*). Profile keywords which can only occur with a single value assignment are disabled for any action by a *#* sign in the action column.

Column A: This column is set to Y(es), if a change of the keyword value will become active immediately. Value N(o) denotes that the value change does not become active until the next start of HCD.

Column Value: You can overwrite the current values. Also, if you position the cursor on a value in this column, you have the following options:

- Pressing PF4 lets you prompt for available values (PF4) where applicable, and select one value from the offered list.
- By pressing PF1, you can obtain an explanation of the selected keyword.

Figure 10 on page 29 shows the contents of a sample profile data set.

Keywords

The following keywords are supported for a profile statement:

Volume serial number to allocate output data sets

To place the IOCP, HCPRIO, and JES3 INISH stream checker data sets to a specific volume within HCD, you can specify this target volume via two profile options:

HCDDECK_VOL

Specifies the volume serial number for allocating a new IOCP, HCPRIO or other data set containing I/O configuration statements.

HCDJES3_VOL

Specifies the volume serial number for allocating a new JES3 initialization stream checker input data set.

In a non-SMS managed environment, the generated corresponding output data set is placed on the indicated volume. The profile option is ignored if the specified output data set already exists on a different volume. In this case, the new output data set replaces the existing data set on this volume. In an SMS-managed environment this option is ignored.

The two options have no effect in an SMS managed environment since SMS overrules the VOLUME parameter.

If the keywords are omitted, the placements of the IOCP, HCPRIO, JES3 INISH, and other I/O configuration data sets are controlled via SMS or ESOTERIC system defaults (ALLOCxx of SYS1.PARMLIB or the UADS, respectively).

Volume serial numbers to allocate log data sets and HCM MCF data set

ACTLOG_VOL

If the dataset names are not managed by SMS, this keyword specifies the up to 6 characters long volume serial number to allocate a new activity log. Using an asterisk (*) indicates that the activity log file will be placed on the same volume where the associated IODF resides.

CHLOG_VOL

If the dataset names are not managed by SMS, this keyword specifies the up to 6 characters long volume serial number where to allocate the change log data set. Using an asterisk (*) indicates that the change log file will be placed on the same volume where the associated IODF resides.

MCF_VOL

If the data set names are not managed by SMS, this keyword specifies the up to 6 characters long volume serial number where to allocate the MCF data set. Using an asterisk (*) indicates that the MCF data set will be placed on the same volume where the associated IODF resides.

Automatic activity logging

CHANGE_LOG

YES/NO. Specifies whether you want to activate change logging (YES). The default is NO.

If enabled, and HCD additionally maintains an activity log file for the IODF, then HCD logs all updates applied to the IODF in a change log file, and automatically generates activity log entries for updates on HCD

objects, for example, add, delete, update or connect, disconnect. These entries are proposals and are shown in the activity log panel where you can modify them before you exit the IODF (see also “Activity logging and change logging” on page 50).

A change of the value setting for this keyword will not be active until the next start of HCD.

Allow or prohibit mixed esoterics

MIXED_ESOTERIC

YES/NO. Specifies whether you want to allow or prohibit mixed devices (DASD and TAPE) under the same esoteric name. If you specify NO, which is the default, and your configuration contains an esoteric with mixed DASD and TAPE devices, the request to build a production IODF will fail with error message CBDA332I. If you specify YES, HCD issues message CBDA332I as warning message and continues the request.

HLQ for exporting IODFs

EXPORTED_HLQ

By default, when exporting an IODF, the generated sequential data set is written with the high-level qualifier (HLQ) of the userID that issued the Export IODF function. If this convention is not suitable for your installation, you can use the keyword EXPORTED_HLQ to specify a different HLQ (up to 8 characters).

Allocation space for data sets allocated due to HCM requests

ALLOC_SPACE

This HCD profile option lets you overwrite the default allocation (CYL,50,50) for data sets that are temporarily allocated in response to HCM requests, such as HCDASMP, HCDRPT, HCDIN. For example, specify: ALLOC_SPACE = HCDASMP,nn where nn is the size (decimal number) used for primary and secondary allocation (in CYL).

Extending allocation space

MCF_EXTENSION

This HCD profile option lets you define additional space when allocating the MCF data sets to allow for updates.

With this keyword, you specify the percentage of additional space that is to be allocated when defining an MCF data set. Per default an MCF data set is allocated with 30 percent additional space than actually needed to hold the MCF data. You can use this space for updates that consume data space without the need to allocate a new MCF and delete the old one. For example, MCF_EXTENSION = 50 allocates 50% additional space.

CHLOG_EXTENSION

This HCD profile option lets you define additional space when allocating the change log data set.

With this keyword, you specify the percentage of additional space that is to be allocated when defining a change log data set. By default, a change log data set is allocated with the same size as the associated IODF. For example, a value set to 50 allocates 50% additional space.

The default extension is 0.

Name and volume serial number for UIM library

UIM_LIBNAME

Specifies the name of the data set containing the UIMs, the associated UDTs, and any help members for the UIMs. If the keyword is omitted, SYS1.NUCLEUS is assumed (Note: only UIMs residing in SYS1.NUCLEUS are read during IPL!).

When UIM_LIBNAME is specified, HCD does not implicitly access SYS1.NUCLEUS for loading the UIMs.

If you specify an asterisk (*) as data set name, HCD assumes that the UIM data set (including SYS1.NUCLEUS) is part of the ISPF load library concatenation chain, contained in the JOBLIB/STEPLIB concatenation chain, or specified in the active LNKSTxx member.

You can only define one data set with the UIM_LIBNAME statement. If you want to specify several data sets, specify an asterisk (*) as data set name and specify the data sets in the JOBLIB/STEPLIB concatenation chain.

UIM_VOLSER

Specifies the volume serial number of the UIM library. Required only if the data set is specified via keyword UIM_LIBNAME and is uncataloged.

Load VM UIMs

VM_UIM

YES/NO. Specifies whether VM UIMs will be loaded. The default is YES. Installations without VM should specify NO to gain some performance improvement during HCD initialization.

Options for text reports

LINES_PER_REPORT_PAGE

Specifies the maximum number of lines per page for reports. The default value is 55.

UPPERCASE_ONLY

YES/NO. Specifies whether all HCD reports will be written in uppercase or not. This is useful when using printers that do not have the English codepage. The default is NO.

Layout of graphical reports

GCR_SCALE

Specifies the scaling factor for graphical reports when using BookMaster. The default is *GCR_SCALE=.6*.

GCR_COMPACT

YES/NO. Allows more objects to be displayed in a graphical report. The default is NO. Depending on the report type, a different maximum number of objects is shown on one page:

Report Type	COMPACT=NO	COMPACT=YES
CU	12 channels, 8 control units	16 channels, 10 control units
LCU	8 channels, 8 control units	8 channels, 8 control units
CHPID	8 channels, 64 control units	16 channels, 64 control units
Switch	1 switch	1 switch
CF	1 coupling facility	1 coupling facility

GCR_FORMAT

Specifies the formatting type:

BOOKIE

For BookMaster. This is the default.

DCF creates a data set for DCF containing script commands.

GML creates a data set for DCF containing GML tags.

GDF creates one or more members in GDF format for printing with GDDM (not for batch).

GCR_FONT

Specifies the font to be used for printing; applicable only if *GCR_FORMAT=DCF* or *GCR_FORMAT=GML* was specified. Specify the appropriate font supported by your installation. For information on how to create a graphical report, see "Create or view graphical configuration reports" on page 257.

Color settings for graphical display function

(The following colors are valid specifications: green, red, blue, black, purple, pink, yellow, brown, and white.)

COLOR_NORM

Specifies the color used when drawing the picture. Make sure that the color is visible on defaulted or specified background. The default is GREEN.

COLOR_TEXT

Specifies the color used for any text in the picture. The default is GREEN.

COLOR_HIGH

Specifies the color to be used when identifying a focused object. The default is RED.

COLOR_BACKGROUND

Specifies the background color. If nothing is specified, the graphical display function uses the standard background of the terminal.

Support of TSO option NOPREFIX

TSO_NOPREFIX

YES/NO. Specifies whether the TSO profile option NOPREFIX is recognized by HCD. The default is NO. If you specify *TSO_NOPREFIX=YES*, and the TSO option NOPREFIX is active, all data set names specified in the HCD dialog are taken as is, i.e. HCD does not add a high-level qualifier. Data set names that are generated by HCD act independent of the TSO option NOPREFIX, and the user ID is added as the high-level qualifier.

Esoteric token when migrating MVSCP input data sets

ESOTERIC_TOKEN

YES/NO. YES specifies that HCD will assign esoteric tokens in ascending order when migrating an MVSCP input data set. NO (which is the default) specifies that no tokens will be assigned.

Control unit type when migrating IOCP input data sets

MAP_CUTYPE

Specifies how a control unit type in an IOCP input data set is mapped to a control unit type in the IODF. Specify one or more of the following mappings:

MAP_CUTYPE = *xxxxx,yyyy-yy*

xxxxx

is the control unit type specified in an IOCP input data set

yyyy-yy

is the control unit type and model to be used in the IODF.

For example parameters, see the sample profile in Figure 10 on page 29.

Extended migration

MIGRATE_EXTENDED

YES/NO. Specify YES to exploit the extended migration possibilities as described in “Changing I/O configurations by editing data sets” on page 291.

If you specify NO (which is the default), the additional keywords are not generated during IOCP build and when re-migrating IOCP input data sets, the migration function ignores the commented *\$HCDC\$ and *\$HCD\$ tags.

Bypass IODF information update for SNA processor

BYPASS_UPD_IODF_FOR_SNA

YES/NO. This provides a possibility of bypassing the attempt to update the IODF information for SNA processors after having successfully built the IOCDS.

If you specify YES, then no attempts will be made to update the IODF with IOCDS status information, which will usually fail as a result of the IODF being in an exclusive access mode with the dialog.

If you specify NO (which is the default), then for SNA and non-SNA processors, an attempt is made to update IOCDS information in the IODF after the IOCDS has been built successfully.

Display information during ACTIVATE

SHOW_IO_CHANGES

YES/NO. This option applies to dynamic activate. When performing both a hardware and software change, specify YES (which is the default) to get information about the channel paths, control units, and devices that are deleted, modified, or added.

Loading an IODF into a data space

IODF_DATA_SPACE

YES/NO. If you specify YES (which is the default), the IODF is loaded into a data space.

If you specify NO, the IODF is loaded into the user address space.

IODF name verification for batch jobs

BATCH_IODF_NAME_CHECK

YES/NO. If you specify YES (which is the default), HCD checks if the IODF specified for a batch job conforms to the naming convention as

described in “IODF naming convention” on page 31. Processing of IODFs with invalid names is limited to deletion.

If you specify NO, HCD does not check the IODF names specified for batch jobs.

IODF checker automation

CHECK_IODF

YES/NO. If you specify YES, HCD checks an IODF for consistency and structural correctness whenever the IODF accessed in update mode is unallocated. This corresponds to the TRACE ID=IODF command and will consume processing time depending on the size of the IODF.

If you specify NO (which is the default), HCD does not check the IODF automatically..

Delay device regrouping

DELAYED_GROUPING

YES/NO. If you specify YES, HCD performs any necessary device regrouping after a device group split only when the IODF is closed. This gives a better response time in the HCD dialog for large IODFs.

If you specify NO (which is the default), HCD performs a necessary device regrouping each time when users exit the *I/O Device List*, or, in case the *I/O Device List* was called from either the *Operating System Configuration List* or the *Channel Subsystem List*, when leaving these lists.

Group devices for activate request

DEVGRP_ACTIVATE

YES/NO. If you specify YES (which is the default), HCD groups device related change requests during dynamic activation. The intent is to reduce memory requirements during dynamic activates. This option is only evaluated starting with HCD 2.2. The default is YES and should only be changed if instructed by IBM service.

Default settings for OS parameters

OS_PARM_DEFAULT

This keyword overrides a parameter default value set by the UIM. The value is used as a default on the HCD Define Device Parameters/Features panel. The syntax is:

```
OS_PARM_DEFAULT = xxxxxx,yyyyyy
```

where:

xxxxxx is the parameter keyword
yyyyyy is the new parameter default value

Example: The LOCANY default value is NO. You can change it to YES by specifying the HCD profile option:

```
OS_PARM_DEFAULT = LOCANY,YES
```

Note: Default values cannot be set differently for different device types. Features, for example SHARED, cannot be defaulted using this keyword.

Extension of the attachable device list of a control unit

CU_ATTACHABLE_DEVICE

This keyword allows the attachable device list of a control unit to be

extended to include additional device types. Both the control unit type and the device type must be defined via UIMs. The value syntax is:

```
xxxxxx,yyyyyy
```

where:

```
xxxxxx is the control unit type  
yyyyyy is the additional device type
```

Note that more than one device type can be added to the same control unit type.

Example:

```
CU_ATTACHABLE_DEVICE = RS6K,3174  
CU_ATTACHABLE_DEVICE = RS6K,3274
```

A change of the value settings for this keyword will not be active until the next start of HCD.

Show partition defaults in IOCP statements

SHOW_IOCP_DEFAULTS

YES/NO. Use this option to write comment lines into the generated IOCP deck which show the partition assignments in effect for those CHPID and IODEVICE statements which make use of the IOCP defaults for the PARTITION and NOTPART values.

Setting this option to YES causes the suppressed PARTITION / NOTPART keywords to be generated as comments prefaced by the tag *\$DFLT\$*.

The default is NO.

Export/import additional configuration objects

SHOW_CONFIG_ALL

YES/NO. Use this option to write additional configuration objects during export of switch configuration statements.

Setting this option to YES affects the *Build I/O configuration data* dialog and batch utility to write configuration statements for unconnected control units and devices in addition to those for switches, if the *Configuration/Switch ID* is specified as *.

The default is NO.

Unconditional generation of D/R site OS configurations

UNCOND_GENERATE_DR0S

YES/NO. If you change a *generated* D/R site OS configuration before building a new production IODF, it loses the *generated* attribute and therefore is not regenerated when the production IODF is built (see also "D/R site OS configurations" on page 78). Instead, you must manually delete the D/R site OS configuration and rebuild the production IODF to get the configuration regenerated automatically.

Setting the UNCOND_GENERATE_DR0S option to YES affects that HCD regenerates D/R site OS configurations whenever a new production IODF is built, independent from whether the configurations have been previously modified or not. This helps to avoid manual user interventions in cases where changes on the primary configuration are not automatically applied to the D/R site OS configuration.

The default is NO.

HMC-wide activation

CONNECTION_TABLE

Use this profile option to specify the name of a data set that contains the connection table for establishing connectivity to HCD on the remote systems via TCP/IP.

RCALL_LOG

YES/NO. Use this profile option to activate logging of remote calls into a data set.

Setting this option to YES allocates a new data set named HLQ.CBDQCLNT.LOG, if it does not yet exist. Otherwise an existing data set is used. The default is NO.

RCALL_TIMEOUT

Use this profile option to set the timeout value for the initial connection to a remote system.

Specify the timeout value in seconds using a decimal number. Specifying zero causes HCD to use the default. The default is 60.

Example

The following figure shows a profile with sample data:

```
/* ***** */
/*
/* HCD Profile
/* Created : 2012-11-27 16:09:15 by user : DOCU
/*
/* ***** */
/* ***** */
/*
/* HCD Profile Section for Standard Profile Options
/*
/*
/* ***** */
ACTLOG_VOL = * /* ACTlog on same volume as IODF */
ALLOC_SPACE = HCDASMP,60
BATCH_IODF_NAME_CHECK = NO /* changed to non-default NO */
BYPASS_UPD_IODF_FOR_SNA = YES /* No IODF update with IOCDs data */
CHANGE_LOG = YES /* CHANGE LOG REQ. ACTIVITY LOG = 0 */
CHLOG_VOL = *
COLOR_NORM = BLACK /* default was GREEN */
COLOR_TEXT = BLUE /* default was GREEN */
CONNECTION_TABLE = DOCU.HCD.CONN(TABLE) /* HMC wide activate */
CU_ATTACHABLE_DEVICE = RS6K,3274 /* extend attachable device list */
ESOTERIC_TOKEN = YES /* Esoteric token: ascending order */
GCR_FONT = X0GT20
HCDDECK_VOL = D83WL2 /* Vol for configuration data set */
HCDJES3_VOL = D83WL4 /* Vol for JES3 output data set */
LINES_PER_REPORT_PAGE = 60 /* Max. number of lines per page */
MAP_CUTYPE = 3705,3745 /* Replace CU type during migration */
MAP_CUTYPE = 3880,3880-23 /* Replace CU type during migration */
MIGRATE_EXTENDED = YES /* Enable migration enhancements */
MIXED_ESOTERIC = YES /* Allow mixed dev. w same esoteric */
OS_PARM_DEFAULT = LOCANY,YES /* Default for parameter LOCANY */
RCALL_TIMEOUT = 10 /* Default is 60 */
SHOW_CONFIG_ALL = YES /* write additional config objects */
SHOW_IOCP_DEFAULTS = YES /* show IOCP default for partitions */
TSO_NOPREFIX = YES /* Enable TSO Noprefix (Default NO) */
UNCOND_GENERATE_DROS = YES /* regenerate D/R site OS config */
***** Bottom of Data *****
```

Figure 10. Example of an HCD Profile

Customizing HCD EXEC procedures

Some of the HCD tasks, invoked from the dialog, generate batch jobs. These batch jobs use EXEC procedures, as shown in Table 2.

Your installation can use normal ISPF or TSO/E facilities to change the job control statements in these EXEC procedures. They are stored in the library SYS1.PROCLIB. You can customize these procedures according to your own needs.

You can also modify the EXEC procedures by using JCL overwrite statements in the HCD dialog. Thus, you can, for example, add a statement that refers to the HCD profile. See “Job statement information used in panels” on page 72 on how to specify JCL statements in the HCD dialog.

Table 2. Batch Jobs Used by the HCD Dialog

HCD Task	EXEC Procedure	Job Step Name	More Details
Build an IOCDS	CBDJIOCP	GO	see “Build an IOCDS or an IOCP input data set” on page 335
Build an IOCP input data set	CBDJIOCP	GO	see “Build an IOCDS or an IOCP input data set” on page 335
Print a configuration report	CBDJRPTS	GO	see “Print configuration reports” on page 343
Compare IODFs and CSS/OS views	CBDJCMPR	GO	see “Compare IODFs or CSS/OS Reports” on page 348
Import an IODF	CBDJIMPT	IMP	see “Import an IODF” on page 352
Transmit part of an IODF	CBDJXMIT	GO	see “Transmit a configuration package” on page 49

Chapter 4. How to work with I/O definition files (IODF)

Overview

This information unit includes:

- IODF naming convention
- Working with I/O definition files (specify, change, create, view, backup, delete, copy, export, import, and upgrade IODFs)
- Working with large IODFs
- Activity Logging
- Using an IODF among different release levels

When you start an HCD session, you need to specify the IODF that HCD is to use. How to do this, how to change to another IODF, and how to use HCD tasks to maintain your IODFs is described subsequently.

Before you can activate your configuration, you must build a production IODF. This task is described in “Build a production IODF” on page 208.

Note: The IODF data sets must be cataloged so that you can use them with HCD.

IODF naming convention

You need to comply to naming conventions for work IODFs, production IODFs, and further data sets associated to an IODF (activity log, change log and HCM MCF data set).

The IODF is a VSAM LINEAR data set with different names for the cluster component and the data component. The name of the data set with a cluster component has the format:

```
'hhhhhhh.IODFcc{.yyyyyyyy. ... .yyyyyyyy}.CLUSTER'
```

The name of the data set with a data component has the format:

```
'hhhhhhh.IODFcc{.yyyyyyyy. ... .yyyyyyyy}'
```

Work IODF

The data set name for a work IODF has the format of:

```
'hhhhhhh.IODFcc{.yyyyyyyy. ... .yyyyyyyy}'
```

hhhhhhh

is the high-level qualifier; up to 8 characters long.

cc is any two hexadecimal characters (that is, 0-9 and A-F).

YYYYYYY

are optional qualifiers, separated by a . and up to 8 characters long. The following qualifiers must not be used as last qualifier: CLUSTER, ACTLOG, CHLOG or MCF.

You can use any number of optional qualifiers but do not make the total name longer than 35 characters because, in some circumstances, HCD appends an additional qualifier.

If you use a change log or an HCM master configuration file, the total IODF name must not exceed 29 characters.

If you omit the high-level qualifier and the enclosing single quotation marks, HCD automatically adds your user prefix (your user ID is the default).

Production IODF

The data set name for a production IODF has the same format as a work IODF. You may specify additional qualifiers to differentiate among IODFs (for example for backup reasons). However, the optional qualifiers must be omitted if the IODF is to be used for IPL or dynamic activation. Thus, the format would be:

```
'hhhhhhh.IODFcc'
```

hhhhhhh

is the high-level qualifier; up to 8 characters long.

cc is any two hexadecimal characters (that is, 0-9 and A-F).

Associated data sets

Files associated to an IODF, if used, also must conform to the IODF naming conventions, plus a required last qualifier:

Activity log (a sequential fixed 80 character data set):

```
'hhhhhhh.IODFcc{.yyyyyyy. ... .yyyyyyy}.ACTLOG'
```

Change log (a VSAM LINEAR data set with cluster component and data component):

```
'hhhhhhh.IODFcc{.yyyyyyy. ... .yyyyyyy}.CHLOG'
```

HCM master configuration file (MCF, a VSAM LINEAR data set with cluster and data components):

```
'hhhhhhh.IODFcc{.yyyyyyy. ... .yyyyyyy}.MCF'
```

Create or specify an IODF

You specify the name of the IODF that you want to use on the *Primary Task Selection* panel. The first time you use the dialog, HCD puts the default name SYS1.IODF00.WORK in the IODF name field (see Figure 27 on page 57). You can type over this name to specify the name you want to use.

If you specify an IODF name that does not exist, HCD assumes that you intend to create a new IODF and displays a panel to let you specify the required attributes. HCD then creates and automatically catalogs the IODF.

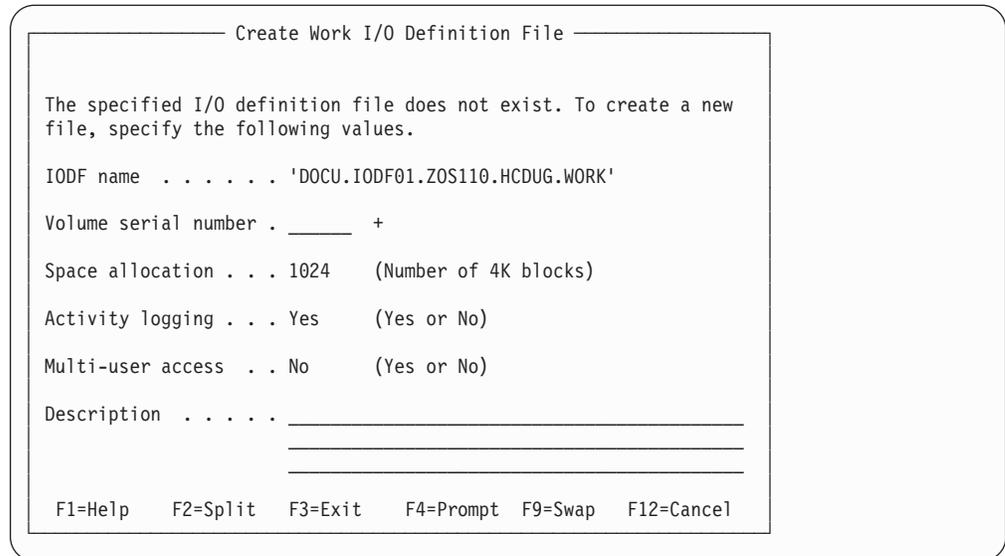


Figure 11. Create Work I/O Definition File

- Volume serial number (of the data volume the IODF will reside on)
This entry is ignored if your system is managed by SMS, otherwise it is mandatory.
- Space allocation
The online HELP gives advice on how much space to allocate. If you run out of space while working with an IODF, you can use the *Copy IODF* task to copy the IODF to a larger data set.
- Activity logging
You have to decide now whether you want HCD to maintain an activity log for the IODF; you cannot specify it later. If you want to use an activity log, your system must have Program Development Facility (PDF) installed.
- Multi-user access
Specify whether or not you want to enable the IODF for multi-user access.
- Description
Here you may enter any useful additional information concerning the IODF, for example, the system it applies to, a special purpose of the IODF, or the author.

The IODF remains in effect throughout all tasks of your current session and later HCD sessions, until you change it.

Multi-user access

Up to z/OS V1R9 HCD, multiple users could read an IODF simultaneously, but no user could read an IODF while it was accessed in update mode by another user. Also, a user could only update an IODF, if no other user accessed the IODF, neither in read nor update mode.

Starting with release z/OS V1R10 HCD, when creating an IODF, you can specify a multi-user access option in the *Create Work I/O Definition File* dialog (Figure 11). The default is single-user access.

Having exclusive access to an IODF, users can also switch between single-user mode and multi-user access using an option in the *Change I/O Definition File Attributes* dialog.

Note:

1. To enable an IODF for multi-user access you need ALTER access authority.
2. You can check whether the multi-user access property is enabled for an IODF using *View I/O definition file information* from the *Maintain I/O Definition Files* task.

With the multi-user access option specified, an IODF is kept in exclusive update mode for a user only for the duration of a single transaction. If the updates of this transaction are committed, another user may update the IODF without requiring the first user to release it.

If a user is updating a multi-user access enabled IODF, HCD implements the following processing: HCD locks the IODF. If multiple subsequent users now also want to apply concurrent updates to the same IODF, they must wait in a queue. However, since the first user's single transaction may last a split-second only, HCD repeats all other users' subsequent update requests a couple of times. If all attempts fail, for example, because the first user updates the IODF using the dialog and is delaying required input, HCD will notify all other requesting users with a message, telling who is currently updating the IODF.

Associated change log files inherit the multi-user access ability from the IODFs.

Note:

When several users simultaneously work on the same IODF, exploiting the multi-user access capability, it is recommended that they co-ordinate their activities in order to ensure the consistency and integrity of the changes made to the IODF.

How to release a lock after an abnormal termination: In most cases of abnormal termination while working with IODFs in multi-user access, HCD invokes a recovery routine that deletes a pending lock. If, however, HCD cannot enter this routine, the lock remains active, and any user, when trying to access the IODF next time, receives a message about who is holding the lock.

In such a case, a user with ALTER access right must re-access the IODF exclusively and select action *Change I/O definition file attributes* to set back the multi-user access capability to NO. This action deletes the lock, and multi-user access can now be reactivated for that IODF.

Sharing IODFs

If you want to share an IODF across two or more systems, you must:

1. Catalog the IODF in the user catalog that is shared by those systems.
2. Define an ALIAS to that catalog in the master catalog of each system that uses the IODF (for details, refer to "Catalog considerations" on page 365).

Note: Control of sharing the IODF resource between multiple systems is achieved via Global Resource Serialization (GRS).

Important:

If you update an IODF simultaneously from different systems that are *not* in the same GRS complex, you may destroy data in the IODF.

Deciding on the number of IODFs

The decision whether to create one IODF for each processor, or to combine the I/O definitions for two or more processors in a single IODF, depends on your environment. This section explains when it is advantageous or even necessary to keep the I/O definitions of two or more processors in the same IODF.

Shared control units and devices

If control units and devices are shared by different processors, the I/O definitions for these processors should be kept in the same IODF to keep change effort to a minimum and to avoid conflicting definitions.

Processor and related OS configuration(s)

For a full dynamic reconfiguration, the IPLed OS configuration must be in the same production IODF as the processor configuration selected for POR.

Coupling facility support

For coupling facility support, you have to maintain your coupling facility definitions for the processors to be connected in the same IODF.

Switch connections

It is recommended that you maintain your switch configurations in the same IODF as the hardware and software configuration definitions to provide complete validation of the data path.

In order to lookup the port connections of a switch, all connected objects to the ports of a switch have to be defined in the same IODF.

CPCs of a z Systems cluster

To manage IODFs, IOCDSs, and IPL parameters within the CPCs of a z Systems cluster from a focal-point HCD, the corresponding processor configurations have to be kept in the same IODF.

Dynamic sysplex reconfiguration

To dynamically reconfigure the I/O configuration of a system within a sysplex from a focal-point HCD, the processor and OS configuration of the sysplex system have to be defined in the same IODF.

CTC connection report

All CTC connections listed in a CTC connection report must be defined in the same IODF. Misconfigurations can be detected only within the scope of one IODF.

Reporting

The scope of the reports (textual or graphical) is a single IODF. All I/O definitions required for a report must be kept in the same IODF.

Validation

The scope of the validation function is a single IODF.

HCM The scope of the configuration shown by HCM is a single IODF.

These requirements and recommendations may lead to a large IODF, depending on the size of the installation. The number of elements in the single IODF may be too large for effective management. HCD provides the possibility of creating manageable *subset IODFs* with a scope limited to a part of the I/O configuration from a *master IODF* describing the entire configuration. For details on this IODF management strategy refer to “The master IODF concept” on page 36.

The master IODF concept

In order to take full advantage of the available HCD functions and to keep a maximum of freedom for reconfigurations it is proposed to keep a master IODF. A master IODF may contain the I/O definitions for an entire enterprise structure. However, a user may decide to divide the I/O configuration definitions of the whole enterprise into several master IODFs, where little or no interference is expected.

Major configuration changes are done in the master IODF. These include:

- processor configurations,
- OS configurations,
- switch configurations, and
- definitions spanning multiple configurations, e. g. coupling facility connections.

After such changes have been made, a production IODF is built. From this IODF comprehensive reports can be obtained.

HCD provides a set of functions which allows the management of very large IODFs for their activation on individual systems, like POR, IPL, dynamic I/O changes, or for z Systems cluster management tasks. Using these functions, subset IODFs containing I/O definitions relevant to only one particular system may be built from the master IODF .

There are no strict rules about what a subset IODF must consist of. Typically it contains:

- a processor configuration with its related OS configuration, or
- all I/O configurations describing the CPCs in a z Systems cluster, or
- all I/O configurations describing the systems of a sysplex.

The content of a subset IODF is specified in a configuration package (see “Work with configuration packages” on page 46).

The subset IODF is transferred to the corresponding target system where it is imported and used as the IODF for that system. A subset IODF constitutes a fully functional IODF. When it is built from a master IODF, the processor tokens are preserved. If necessary, updates concerning the target system alone may be carried out using the subset IODF. Subsequently, the subset IODF can be sent back to the system administering the master IODF and merged back into the master IODF, thereby updating it with the changes made at the target system.

Figure 12 on page 37 illustrates the possible flow of I/O information according to the master IODF concept.

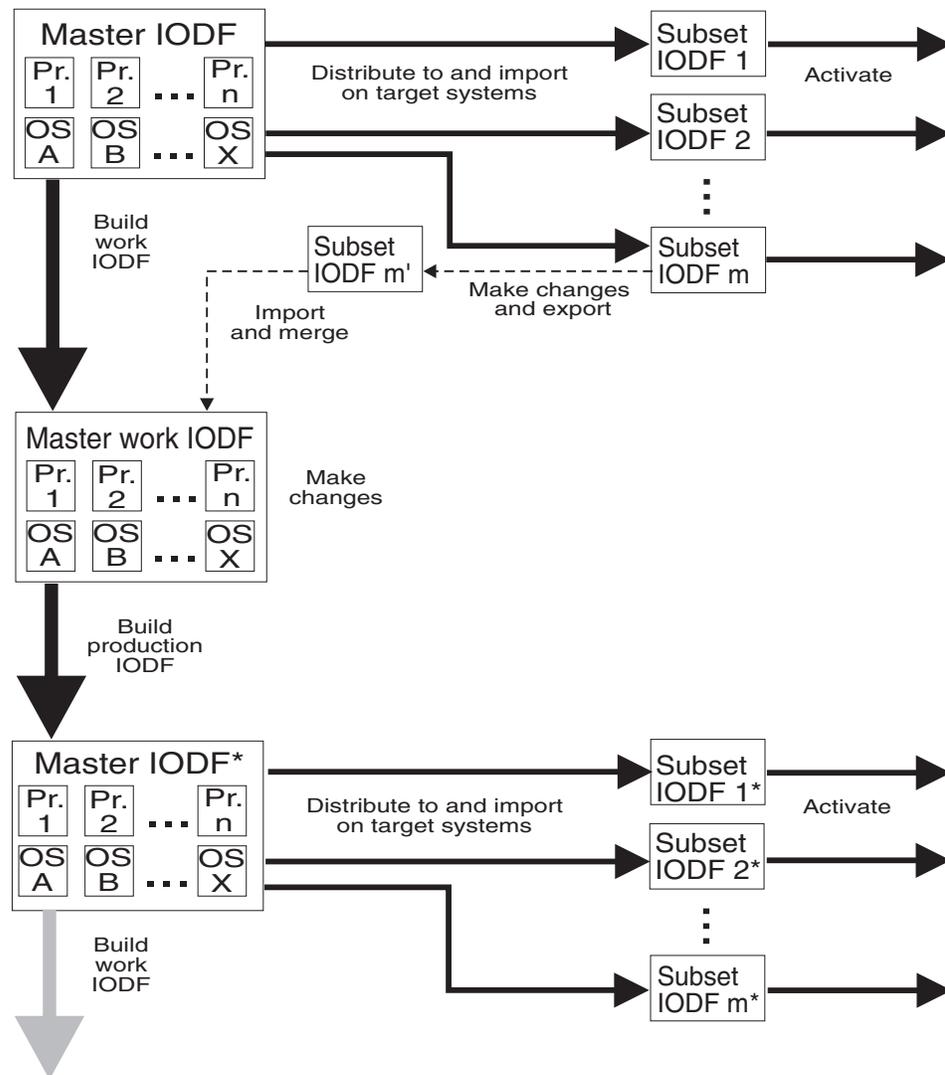


Figure 12. IODF Distribution/Merge Process

How HCD arranges devices into groups in an IODF

IODF versions up to V4 contained a separate device definition record for each single device. To reduce the size of IODFs and to improve the processing performance of large configurations, z/OS V1.7 HCD uses a new IODF format V5, arranging single devices into a device group, if they have the following characteristics in common:

- The device numbers of all devices are in consecutive sequence.
- All devices of the group have the same device type (unit, model) and attribute values (Serial-#, Description, VOLSER).
- All devices of the group are attached to the same control unit(s).
- All devices of the group are connected to the same processors/channel subsystems and have the same corresponding processor-specific attributes.
- All devices of the group are connected to the same operating systems and have the same corresponding OS-specific attributes (device type, parameters, features, console definition, subchannel set number).

- For each OS and each eligible device table (EDT) in the OS, all devices of the group are connected to the same esoterics.

Devices that adhere to these rules are aggregated into device groups containing the maximum number of applicable devices. If you apply a change on one or more devices from a group, HCD checks how to rearrange the devices and device groups contained in the IODF in order to achieve the best possible organization of devices into groups again according to the specified rules. Another example may be that you change the attributes of a group in a way that two previously similar groups now must be merged into one group. However, the required rearrangements are not necessarily performed by HCD at once, but may be delayed to an appropriate point in time.

Change to another IODF

You can work with only one IODF at a time. If, during an HCD session, you want to change to another IODF, you must return to the *Primary Task Selection* panel and specify the new IODF. If the old IODF has an activity log and has been modified, a panel is displayed to let you add a comment into the activity log (“Activity logging and change logging” on page 50 tells you how). The new IODF now becomes the current IODF.

Change a production IODF / Create a work IODF based on a production IODF

HCD allows you to perform all tasks on the data stored in a production IODF as long as you do not try to change it. If you try though, HCD displays the *Create Work I/O Definition File* panel where you can define a new work IODF based on the current production IODF.

HCD then copies the production IODF to that new work IODF, makes the work IODF the currently accessed IODF, and applies to it all further changes.

You must specify a new data set for the work IODF; you cannot use an existing one. HCD creates a default work IODF name by appending the qualifier WORK to the production IODF name. You can change this default work IODF name.

When you have completed the changes, you can use HCD to build a new production IODF from the work IODF.

View active IODF

HCD provides information about the IODF that has been used for last POR/IPL or for dynamic activation (that is, the currently active IODF); in addition the operating system ID and EDT ID used for IPL are shown, and the configuration token that is currently active in the HSA (hardware system area). For a description of this function, see “View active configuration” on page 230.

Backup work or production IODFs

You can use the *Backup* action bar choice on any action list panel to copy a work or production IODF to a backup data set. Thus you can keep track of different stages of the configuration as well as retrieve data lost by accident. When you do the first backup, you must also specify the volume serial number, if applicable, and the space allocation for the backup data set. HCD uses the normal catalog process to catalog copies of the IODFs.

If you backup a work IODF, you need to specify the name of the backup data set only once for each IODF (when you do the first backup). HCD saves the name of the backup IODF data set, so you can reuse this data set for each subsequent backup (or use a different one if you want).

If you backup a production IODF, HCD does not save the name of the backup data set, because a production IODF cannot be edited. It is suggested that you maintain a backup copy of your production IODF on a separate volume that is accessible from all systems that will be sharing the backup. When the primary IODF volume is inaccessible or the IODF itself is corrupted, the system can be IPLed through a backup IODF on the alternate volume.

It is also recommended that you choose an alternate high level qualifier for your backup IODF since a lost IODF volume may imply a lost IODF catalog. This high level qualifier can be cataloged in either the master catalog or in an alternate user catalog.

As an alternative method to create a backup IODF, you can use the following procedure:

1. Select *Maintain I/O definition files* from the *Primary Task Selection* panel (Figure 9 on page 18).
2. Select *Copy I/O definition file* from the *Maintain I/O definition Files* panel (Figure 13).
3. Specify the name of the backup IODF.

Maintain IODFs

HCD provides the tasks listed on the Maintain I/O Definition Files panel (Figure 13) to help you maintain your IODFs. You can reach this panel from the *Primary Task Selection* panel (see Figure 27 on page 57).

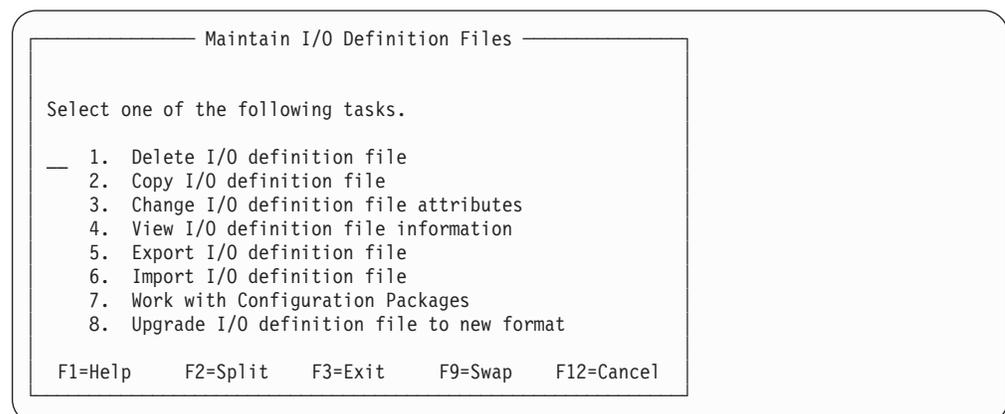


Figure 13. Maintain I/O Definition Files

Delete an IODF

This task deletes an IODF. If the IODF has an activity log, that log is also deleted. Also, if an HCM master configuration file (MCF) is associated with the IODF, it is deleted along with the IODF. HCD asks you for confirmation before actually deleting the IODF. It will not be possible to delete the currently active IODF of the system, HCD is running on.

Copy an IODF

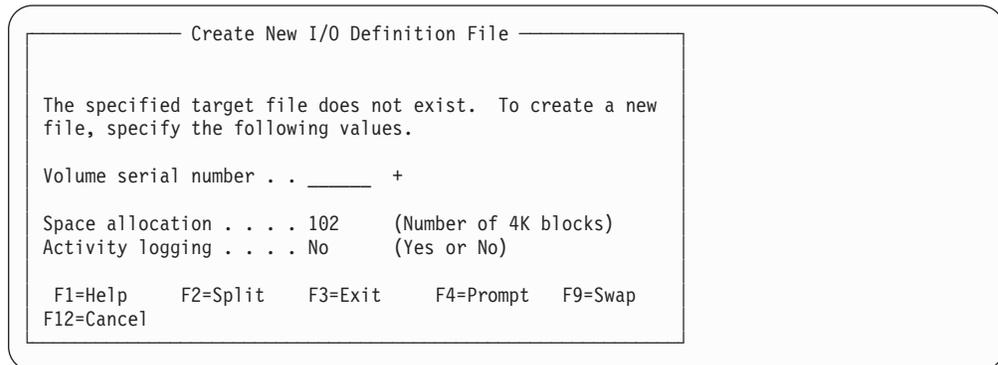
You can invoke the task *Copy I/O definition file from the Maintain I/O Definition Files panel* (Figure 13 on page 39). This task copies any IODF to another IODF (either existing or new). You must specify the name, volume serial number, if applicable, and the space allocation of the target data set. If the IODF has an activity log, that log may also be copied. Also, if an HCM master configuration file (MCF) is associated with the IODF, it is copied along with the IODF. However, a change log file (CHLOG), if available, is not copied.

During a definition task, you can use *Copy IODF* to copy the existing data to a larger data set if you have allocated insufficient space to a work IODF. In this case, you need to return to the *Primary Task Selection* panel afterwards to specify the new data set as the IODF you are working with from now on.

You can also use the *Copy I/O definition file* task to upgrade a V4 IODF to a V5 IODF. The result of the copy process will always be a V5 IODF.

In the *Copy I/O Definition File* dialog, if you specify a target IODF that does not yet exist, HCD displays the dialog from Figure 14 where you can create a new target IODF. The space allocation default depends on the source IODF:

- for a V5 source IODF, the allocation default is the number of **allocated** blocks of the source IODF.
- for a V4 source IODF, the allocation default is the number of **used** blocks of the source IODF.



```
----- Create New I/O Definition File -----  
  
The specified target file does not exist. To create a new  
file, specify the following values.  
  
Volume serial number . . _____ +  
Space allocation . . . . 102      (Number of 4K blocks)  
Activity logging . . . . No       (Yes or No)  
  
F1=Help   F2=Split   F3=Exit   F4=Prompt   F9=Swap  
F12=Cancel
```

Figure 14. Create New I/O Definition File

If you copy an IODF which is enabled for multi-user access, this property is not inherited by a target IODF. However, an existing target IODF defined with the multi-user access property will always preserve this property, independent from the source IODF.

You can also invoke the *Copy I/O definition file* task in batch mode. For details refer to “Copy IODF” on page 342.

Change IODF attributes

With this task, you can change certain attributes of an IODF. You can change the description and you can enable or disable the IODF for multi-user access.

Note: To enable or disable an IODF for multi-user access you need ALTER access authority.

View an IODF

This task displays information about the currently accessed IODF. The information includes the type and version of the IODF, its description, the creation date, the last update, and how much of the allocated space for the IODF data set has been used.

You can also invoke the *View IODF* task from the *View* action bar, and by issuing the SHOWIODF command from the command line.

```
View I/O Definition File Information

IODF name . . . . . : 'IODFST.IODFD0'
IODF type . . . . . : Production
IODF version . . . . . : 5

Creation date . . . . . : 2004-10-14
Last update . . . . . : 2004-10-21 14:48

Volume serial number . . : CMNSTC
Allocated space . . . . . : 2973 (Number of 4K blocks)
Used space . . . . . : 2965 (Number of 4K blocks)
  thereof utilized (%) 96

Activity logging . . . . . : Yes
Multi-user access . . . . . : No
Backup IODF name . . . . . :

Description . . . . . : IODF for raised floor 710
                        For system D0, D2, D4 and D6
                        incl. sensed data and mig. SW data

ENTER to continue.

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

Figure 15. View IODF information

Export an IODF

This task sends an IODF, and optionally, its activity log file to another (local or remote) system. On the Export IODF panel, specify or revise the IODF name you want to export, the user ID, or the nickname (only for an attended target system), node ID, and status (attended or unattended) of the operating system (OS) to which the IODF is to be sent.

If the target system is unattended, the IODF is sent as a job to the target system, which must be a system of type MVS. In this case no explicit action on the target system is required. You can specify whether to replace an existing IODF with the same name. If the existing IODF is the active IODF for the remote system HCD is running on, replace will not be possible.

If the target system is attended (receiving to be done by the user on the remote system), the IODF is sent to the target system as a sequential data set. There it has to be received by using the TSO RECEIVE command. As a second step the IODF is imported on the target system (see also “Import an IODF” on page 44).

To export an IODF, HCD uses the TSO command TRANSMIT. Therefore, HCD creates a cataloged sequential data set named *tsoid.EXPORTED.iodefname*, where *tsoid* is the sending TSO user ID or the TSO prefix, or is determined by the profile option *EXPORTED_HLQ*, and *iodefname* is the part of the IODF data set name after the high-level qualifier.

After processing the TRANSMIT command, the sequential data set is deleted.

If you have specified to send the selected IODF to a system with an operating system running in unattended mode, use the panel from Figure 16 to define the job control language (JCL) statements for importing the IODF on that unattended system.

Define JCL for IODF Import

Specify or revise the job control statements for importing the IODF on an MVS system.

```
//JOBNAME JOB (ACCT,BOX),'USER',CLASS=CLASS,
//          MSGCLASS=MSGCLASS,MSGLEVEL=(1,1),
/*LEAVE THIS JOB CARD UNCHANGED: USERID AND PASSWORD ARE GENERATED
/*ROUTE   XEQ HCD3
```

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

Figure 16. Define JCL for IODF Import

You can also invoke the *Export IODF* task by using the *Export* task in batch mode. For details refer to “Export an IODF” on page 353.

Prerequisites:

1. Network Job Entry (NJE) must be active.
2. The target user ID and password, and, if the target system is not SMS managed, the volume serial number of the receiving data volumes have to be known when the IODF is exported to an unattended system.

User authentication for unattended mode

If you export an IODF to a target system running in unattended mode, there are two alternatives for user authentication:

- sending the target user ID and password to the target node (that is, the user ID which receives the IODF)
- defining a surrogate user ID on the target to act on behalf of the receiver

Sending user ID and password:

- The first statement controls whether jobs coming from node_s are allowed to enter the system from node_s. It also controls, whether jobs that enter the system from node_s nodes have to pass user identification and password verification checks.
- The second statement defines a user_r.SUBMIT profile in the SURROGAT general resource class for user_r who requires a surrogate user to act on his behalf.
- The third statement authorizes user_s to act as a surrogate for user_r.

Import an IODF

This task imports previously received IODF data into HCD. You can export and import IODFs between different HCD versions. You can, for example, export an IODF from HCD 1.4 and import it with HCD 1.7. Note that in this case you have to upgrade the earlier version IODF before using it with HCD, because the export/import function does not change the format of the IODF.

You can also invoke the *Import IODF* task by using the *Import* task in batch mode. For details refer to “Import an IODF” on page 352.

At an unattended target system, the IMPORT batch utility is invoked automatically when an IODF, with its associated job control, arrives at the system.

Upgrade an IODF

This task upgrades an IODF from a back-level format to the new format that is required for the current release of HCD.

Invoke the *Upgrade I/O definition file to new format* task as follows:

1. On the HCD *Primary Task Selection* panel (Figure 9 on page 18), specify the IODF to be upgraded at the bottom of the menu and then select option 6. *Maintain I/O definition files.*
2. From the *Maintain I/O Definition Files* panel (Figure 13 on page 39), invoke option 8. *Upgrade I/O definition file to new format.*

HCD displays the following dialog:

```

Upgrade I/O Definition File

Select a target for the IODF to be upgraded.

IODF name . . . . . : 'DOCU.IODFA0.WORK'

Target of upgrade . . . . 1  1. To new work IODF
                           2. In place

Condense IODF . . . . . 1  1. Yes
                           2. No

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

```

Figure 18. Upgrade an IODF

Table 3 on page 45 shows the options you have. You can either:

- Upgrade into a new work IODF
- Upgrade in place

Table 3. Size considerations when upgrading a back-level IODF

Upgrade IODF		Upgrade into New IODF	Upgrade in Place
From	To		
V4 IODF	V5 IODF	<ol style="list-style-type: none"> 1. with Condense option: Default size of new IODF is double the size of the related production IODF. You can change this default size, if necessary. 2. without Condense option: Default size of new IODF is the size of V4 IODF. 	V5 IODF has same size (number of allocated blocks) as V4 IODF

Note:

1. The *Upgrade in place* option cannot be requested for a production IODF (as a production IODF cannot be altered). The final result of the upgrade IODF function is always a work IODF.
2. If you plan to add or change many devices in the configuration, ensure that you increase the space allocation when creating the IODF to allow for these changes.

If you select to upgrade in place, the accessed IODF is formatted to a V5 IODF. While the allocated and used space values do not change, the space utilization of the used blocks will decrease depending on the number of devices that can be grouped.

If you select to upgrade to a new IODF, the accessed IODF will not be changed. HCD displays the **Create Work I/O Definition File** dialog (see Figure 19 on page 46). The space allocation default is set as follows:

- If you select Condense IODF = Yes, the default allocation value is set to double the size required for a corresponding production IODF. This may result in much lower space requirements than without condensing.
- If you select Condense IODF = No, the default allocation value is set to the number of used blocks of the IODF to be upgraded.

```

----- Create Work I/O Definition File -----

The current IODF was created by an earlier HCD release, and you
have requested upgrade to a new data set. To create this data
set, specify the following values.

Source IODF . . . . : 'DOCU.IODF00.TREXDOCU.HCMUG'
  Allocated space . . : 1024
  Used space . . . . : 27

IODF name . . . . . _____

Volume serial number . _____ +

Space allocation . . . 64      (Number of 4K blocks)

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

```

Figure 19. Create Work I/O Definition File

After the upgrade completed successfully, HCD issues a message of success and accesses the upgraded IODF with which you can start to work now.

You can also invoke the *Upgrade I/O definition file to new format* task in batch mode. For details refer to “Upgrade IODF” on page 330.

Note: An IODF can also be upgraded using the *Copy IODF* task (see “Copy an IODF” on page 40).

Work with configuration packages

Configuration packages define subset IODFs which are extracted from the accessed IODF (centrally administered master IODF) and distributed for activation at selected target systems, while keeping the processor token in sync.

When invoked with option *Work with Configuration Packages* from the panel shown in Figure 13 on page 39, this task displays the Configuration Package list. Configuration packages can be added and edited, transmitted from a production IODF, and they can be deleted.

```

Goto Backup Query Help
-----
Configuration Package List Row 1 of 2 More:      >
Command ==> _____ Scroll ==> PAGE

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

IODF name . . . . : IODFST.IODF11.MASTER

Package  -----Target----- ----Last sent---
/ Name   User   Node   Date   Time Description
_ CB88   OS390H1  PKSTCB88  1997-10-22  06:28 For A/T system M88TSO
_ CB89   OS390H1  PKSTCB88  1997-10-21  04:34 For A/T system M89TSO
***** Bottom of data *****

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

```

Figure 20. Sample Configuration Package List — left panel

The target user and node as well as the description of a configuration package can be edited by overwriting the information on the panel. When scrolling to the right you can also overwrite the target IODF name and attended/unattended information.

Define a configuration package

To define a new configuration package perform the following steps:

1. On the Configuration Package List, use F11=Add. If you want to use an existing configuration package as a model, select the package and the *Add like* action from the context menu (or use action code **a**). The Add Configuration Package panel is displayed.

```

Add Configuration Package

Specify the following values.
Package name . . . . . CB8A
Package description . . For A/T system M8ATSO

User ID . . . . . OS390H1
Node ID . . . . . PKSTCB88

Operating system status 2 1. Attended
                        2. Unattended (MVS only)

Target IODF name . . . . 'IODFST.IODF8A'
Target volume . . . . . CMNSTC (for unattended, if not SMS)

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

Figure 21. Add Configuration Package

2. Enter a package name and other entry data as appropriate. The user ID and node specify the destination of the corresponding subset IODF. The name of the accessed IODF is used as default for the name of the IODF at the target system and for the Descriptor field 1 and Descriptor field 2 parameters during the Build Production IODF step, when the configuration package is transmitted (see step 5 on page 50 in section “Transmit a configuration package” on page 49). If no value is entered for the Operating system status, Attended is assumed which means that a user has to import the subset IODF into HCD at the target system. This is done automatically if 2 (Unattended) is specified.
3. After you press ENTER, HCD displays the updated Configuration Package list.

Edit a configuration package

The target user and node, the attended/unattended information, the target IODF name, and the description of a configuration package can be edited on the Configuration Package List (see Figure 20 on page 46).

To edit a configuration package perform the following steps:

1. Display the Configuration Package List.
2. Make changes to the configuration package by overwriting the entries on the list panel.
To change the IODF name and attended/unattended information scroll to the right and overwrite.
3. Press ENTER.

Delete a configuration package

To delete a configuration package perform the following steps:

1. On the Configuration Package List select the package and the *Delete* action from the context menu (or use action code **d**). The Confirm Delete Configuration Package panel is displayed.
2. Press the Enter key to confirm the deletion. HCD displays the updated Configuration Package list.

Work with configuration package objects

Configuration package objects are operating systems or processors. The objects in a configuration package determine the scope of the corresponding subset IODF. To change the content of a configuration package select the package from the Configuration Package List (see “Work with configuration packages” on page 46) and the *Work with Configuration Package Objects* action from the context menu (or use action code **s**). The Configuration Package Objects List is displayed.

Add a configuration package object: To add a configuration package object to a configuration package perform the following steps:

1. Use F11=Add or select an object from the Configuration Package Object List and the *Add like* action from the context menu (or use action code **a**). The Add Configuration Package Object panel is displayed.
2. Select the Configuration type, processor (PR) or OS configuration (OS), and specify the Configuration ID of the object.
3. Press the Enter key. The updated Configuration Package Object List is displayed.

Merging changes into a master work IODF: If a system has been supplied with a subset IODF and configuration changes have been made using the subset IODF, you may want to update the master IODF with these changes. This can be done by merging the changed configuration package objects from the subset IODF back into the master work IODF.

First the updated subset IODF is exported on the corresponding system and imported at the system administering the master IODF. You can update the master work IODF by merging or by replacing configuration package objects.

The *Merge* action updates new/changed data in the master work IODF.

The *Replace* action first deletes the object configuration present in the master work IODF and then replaces it with the new one.

To merge a configuration package object into a master IODF perform the following steps:

1. Access the master work IODF the configuration package belongs to.
2. Display the Configuration Package Object List and select one or more objects. Select *Merge* (action code **m**) or *Replace* (action code **r**) from the context menu.
3. Specify the source IODF the configuration package refers to and press the Enter key.
4. The master work IODF is updated and a new master production IODF can be built.

Note:

1. Switch configurations are not repeated by *Merge* and *Replace*. If necessary, switch configuration changes should be made in the master IODF before *Merge* or *Replace* are performed.
2. The *Merge* and *Replace* actions utilize the *Repeat* action for entering the new configuration data into the master work IODF. Refer to “Repeating (copying) processors” on page 89 and “Repeating (copying) operating system configurations” on page 79 for details on the *Repeat* action.

Delete configuration package objects: To delete a configuration package object from a configuration package perform the following steps:

1. Select an object from the Configuration Package Object List and the *Delete* action from the context menu (or use action code **d**).
2. Press the Enter key. The updated Configuration Package Object List is displayed.

Transmit a configuration package

Transmitting a configuration package means building a subset IODF and transmitting it to the target system specified in the configuration package. This action can only be carried out from a production IODF. To transmit a configuration package perform the following steps:

1. Select the package from the Configuration Package List and the *Transmit configuration package* action from the context menu (or use action code **x**).
2. If the following conditions are fulfilled, the Transmit Configuration Package panel is displayed:
 - The accessed IODF is a production IODF.
 - The selected configuration package contains at least one configuration object (OS or processor).
 - In the selected configuration package a destination user and node are specified.
 - In the selected configuration package a target IODF is specified.

Descriptor field 1 is defaulted to the HLQ of the target IODF name in the configuration package; Descriptor field 2 is defaulted to the 2nd qualifier of the target IODF name in the configuration package.

```

Transmit Configuration Package

Package name . . . . . : PACK1          Package for BOEHCD1
High level qualifier . . HCDI          Volume . . . . . _____ +
JCL member used . . . . . CBDJXMIT     Space . . . . . 32 (4K blocks)

Descriptor field 1 . . . WEID          Descriptor field 2 IODFCC
Target user ID . . . . . WEID         Target node ID . . . BOEHCD1
Operating system status 1 1. Attended
                             2. Unattended (MVS only)

Specify or revise the job control statements for the transmit job.

//HCDIXMT JOB (3243),'WEID',CLASS=A,MSGCLASS=H,REGION=0M,NOTIFY=HCDI
//*
//*
```

Figure 22. Transmit Configuration Package

3. Review the entry data displayed. You may change Descriptor field 1, Descriptor field 2, the target user ID and node ID, the Operating system

status, the high level qualifier and/or the volume of the IODFs and other data sets to be generated temporarily, as well as the estimated size of the target IODF.

If you want to use a customized transmit procedure you can specify a different JCL member and/or a JCLLIB parameter in the job control statement area. Once changed, the statements will be retained across sessions. If you want to use different load libraries, specify JOBLIB, because the procedure contains several steps.

If the transfer is performed unattended, the *Specify Target IODF and User Password* dialog is displayed. Refer to “User authentication for unattended mode” on page 42 for more information.

The master production IODF is updated with the last sent date and time when the job stream is built. The JCL member provided, CBDJXMIT, consists of the following steps:

1. A temporary work and a temporary production IODF, according to the values entered in the Transmit Configuration Package panel are initialized.
2. The processor configurations included in the selected configuration package are written to a data set and then migrated to the temporary work IODF.

To generate coupling facility connections, HCD needs both the CF sender and CF receiver channel paths, or peer channel definitions, within the same IODF. Therefore, if a processor of the configuration package contains a connected sending CF channel path, the processor containing the coupling facility partition will be included (with the coupling facility partition only) even if it is not part of the configuration package. (The receiving CF channel paths of the CF partition are indicated as occupied if they have connections to processors outside of the package.)

The switches and ports which contain connections to a processor of the configuration package are distributed as well. Ports that contain connections to a processor, switch or control unit outside of the scope of the configuration package are indicated as occupied.

3. The OS configurations related to the selected configuration package are written to a data set and then migrated to the temporary work IODF.
4. The switch configurations related to the selected configuration package are written to a data set and then migrated to the temporary work IODF.
5. A temporary production IODF is built from the temporary work IODF.
The processor token is not changed by this action.
6. The production IODF is exported to the specified user ID and node, attended or unattended, as selected.
7. The temporary work and production IODFs are deleted.

The transmit action can also be carried out using a batch utility. A sample job, CBDSXMIT, has been provided in SYS1.SAMPLIB.

Activity logging and change logging

When you end an HCD session or access a different IODF after modifying some configuration data, and if an activity log is enabled for the currently accessed IODF (“Create or specify an IODF” on page 32 explains how to specify an activity log), then HCD displays an activity log panel, showing the information which will be added to the activity log.

```

Date & Time . . . . . : 2005-11-07 10:54:46
User . . . . . : BOKA
I/O definition file . . : 'BOKA.IODF00.ACT'
Change reference number : 000002

***** ***** Top of Data *****
000001 Type your log entries here ...
000002 ...
000003 ...
000004 ...
***** ***** Bottom of Data *****

```

Figure 23. Example of an Activity Log without automatic logging

This information comprises the date and time, the user ID of the user who modified the IODF, the name of that IODF, and a change reference number. You can add your own comments to the log, for example, describing what you have done.

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

```
CHANGE_LOG = YES
```

With this setting, HCD generates automatic entries into the activity log panel, describing the updates on HCD objects, like for example, add, delete, or connect, disconnect. You can see examples of such entries in Figure 24 on page 52. You can edit the entries before you exit this panel.

The activity log panel is an ISPF/PDF panel, so the normal ISPF/PDF rules apply to it. Activity log editing requires the profile option AUTOSAVE ON. When calling the ISPF editor, macro CBDCACTL is used. You can tailor this macro to your installation needs (for example, for setting specific profile options). Use the F3=Exit key to continue.

HCD appends the qualifier ACTLOG to the IODF data set name to create the data set name for the activity log. If the ACTLOG data set does not yet exist, HCD dynamically allocates one using ESOTERIC system defaults (see “SMS-related considerations” on page 366). If you want to use a specific volume, you can specify the volume serial number to allocate a new activity log in the HCD profile (see “Defining an HCD profile” on page 19).

You can view or print the activity log associated with the currently accessed IODF during an HCD session by selecting the option *Print or compare configuration data* from the *Primary Task Selection* panel and then *View the activity log* or *Print the activity log*.

You can also use the ISPF/PDF facilities to browse or print an activity log.

HCD generates its proposed activity log entries from the change log file. Setting the CHANGE_LOG = YES option in the HCD profile causes HCD to create the change log file and store in it all update operations on the related IODF in a wrap-around manner.

The change log file is a VSAM data set. If it is not accessible for any reason, HCD cannot create any automatic activity log entries.

The name of the change log file is built from the related IODF data set name plus the suffix CHLOG. It is allocated in the same size as the related IODF.

You can write the contents of the change log file to the HCD trace data set using the TRACE command with ID=CLOG: trace on, id=clog, level=8

With the profile options CHLOG_VOL and ACTLOG_VOL, in non SMS-managed environments, you can specify the volume serial numbers where to allocate a new change log or activity log.

```

View Activity Log
-----
Command ==>                               Scroll ==> PAGE
Activity log . . : BOKA.IODF00.ACT.ACTLOG

***** Top of Data *****
REFERENCE  DATE          TIME          USER ID          IODF NAME
-----
000001:    2005-10-13  17:02:15  DOCU             DOCU.IODFA0.TES
          Device 0001 added
          OS device MVS1.0001 added
          Device 0001(2) added
          OS device MVS1.0001(2) added
000002:    2005-10-13  17:14:11  DOCU             DOCU.IODFA0.TES
          Processor PROC2 added
          CSS PROC2.0(2) added
          Device 0003 added
          OS device MVS1.0003 added

***** Bottom of Data *****

F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND
F6=RCHANGE   F7=UP        F8=DOWN      F9=SWAP      F10=LEFT
F11=RIGHT    F12=RETRIEVE

```

Figure 24. View Activity Log

Rules for automatic activity logging

HCD applies the following rules during automatic activity logging:

- When creating an object with immediate updates without leaving HCD, HCD only creates an *'add object'* log entry in the activity log.
- When updating the same object multiple times during one HCD session, HCD creates only one *update object* entry.
- Consecutive IDs, for example for channel subsystems, channel paths, control units or devices, are marked with a 'quantity' number in parenthesis behind the starting ID.
- When repeating an object into a new IODF, no entry is created in the target IODF since this IODF is not in access by the user.

Actions performed on IODFs and related activity and change log files

- When you delete an IODF, HCD also deletes the related activity and change log files.
- When you copy an IODF, HCD also copies the activity log file with all its contents. However, HCD does not copy the change log file. Instead, a new change log file is created for the new IODF.
- When you build a production IODF, HCD copies the activity log file, but not the change log file.
- When you enable or disable an IODF for multi-user access, the same action is applied to the change log file. Before an update request is performed on an

IODF, the change log file is refreshed to have the latest updates available. The activities of multiple users are logged in chronological order.

- An activity log data set is not enabled for multi-user access. If multiple users simultaneously access and update the related IODF, the activity log file is accessed sequentially in the order the users end the HCD session or access a different IODF. The users' activity log entries are written to the data set grouped by the user ID.

IODF release level compatibility

If you plan to share an IODF among multiple z/OS or OS/390 systems that are at different release levels, you have to consider several restrictions concerning IPL, IODF usage, and dynamic reconfiguration.

Table 4 shows possible z/OS HCD levels that could be installed on your system. For each HCD level, the table shows whether you can take the following actions:

IPL IPL the z/OS release

HCD Use HCD to work with the IODF

Dyn Do a dynamic reconfiguration

Table 4. Coexistence Considerations for IODFs

HCD installed with:	Action	IODF Level	
		V4 IODF	V5 IODF
z/OS 1.9 HCD (HCS7740)	IPL	Yes	Yes
	HCD	Yes (See note 1.)	Yes (See note 2.)
	Dyn	Yes	Yes
z/OS 1.10 HCD (HCS7750) or later	IPL	Yes	Yes
	HCD	Yes (6)	Yes
	Dyn	Yes	Yes

The following notes refer to Table 4.

Note:

1. Read-only access is provided. Upgrade of V4 IODF to V5 IODF is required for updates.
2. With HCD releases earlier than z/OS 1.10, you cannot directly access a multi-user access enabled V5 IODF. First you must remove the multi-user access capability from the IODF with z/OS 1.10 HCD or higher. In addition, you need to install the coexistence PTF for the corresponding z/OS release to prevent earlier HCD releases from accessing an IODF prepared for multi-user access (APAR OA22842).

Chapter 5. How to use the dialog

Overview

This information unit explains the following topics:

- Window layout
- Working with lists
- Promptable fields
- Commands and function keys
- Getting help
- Navigating through the dialog
- Filtering
- Job statement information used in dialogs

Window layout

Figure 25 explains the areas of an HCD dialog window. These areas appear in the same position on every window unless you use ISPF to change the position of the command line. Not all areas are included on all dialogs.

```
Goto Filter Backup Query Help 1
-----
                                I/O Device List      Row 1 of 14  More:
Command ==> _____ Scroll ==> PAGE 2

Select one or more devices, then press Enter. To add, use F11. 3

-----Device----- --#--- -----Control Unit Numbers + -----
/ Number  Type +      CSS OS 1--- 2--- 3--- 4--- 5--- 6--- 7--- 8---
- 0001    3278-3    3  2  00E1 _____
- 0002    3279-2    3  4  0097 _____
- 0098    9033      2    0098 _____
- 0099    9032      2    0099 _____
- 00C1    3480      2  1  00C1 _____
- 01D1    3390      2  1  00D1 _____
- 01D2    3390      2    00D1 _____
- 01D3    3390      2    00D1 _____
- 01D4    3390      2  1  00D1 _____
# FFFC    CFS          FFFE _____
# FFFD    CFS          FFFE _____
# FFFE    CFS          FFFE _____
# FFFF    CFS          FFFE _____

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

Figure 25. Example of a window layout

- 1** Action bar.
HCD provides an action bar-driven interface. You can select any of the action bar choices and display pull-down choices. Use F10=Actions to move the cursor to the action bar.
- 2** Title Line.
 - Shows the window title.
 - Displays window identifiers, if you have asked for them.

- Displays positional information for the work area:
Row 1 of 3302 specifies that the first row of data that is displayed in the work area is the first row of 3302 available rows.
More: > specifies that more data can be seen by moving the work area to the right using F20=Right. This can also be More: <, which means moving is possible to the left, or More: < >, which means moving is possible to the left and right.
Instead of Row 1 of 3302, Filter Mode would be shown if the action list is filtered. For additional information, refer to “Filtering” on page 69.

3 Instruction Area.

Tells you how to proceed on the dialog window. On action lists (see “Action lists” on page 59) you can get more instruction information by pressing the F13=Instruct key or by using the INSTRUCT command.

4 Work Area.

You can use the following facilities to control the work area:

- The *Filter* action reduces the number of objects in an object list; this is described further in “Filtering” on page 69.
- The F7=Backward and F8=Forward keys scroll the work area backward and forward. The end of the information is indicated by a line containing - end - or BOTTOM OF DATA.
- The F19=Left and F20=Right keys move the work area of an action list left and right (or RIGHT and LEFT command).
- The LOCATE command scrolls an object list so that a specific object (or the nearest lower match) is at the top of the work area. Search criteria is the object identifier shown in the leftmost data column.
- The TOP command scrolls backward to the top of the work area.
- The BOTTOM command scrolls forward to the bottom of the work area.

Figure 26 shows the areas of a data entry dialog.

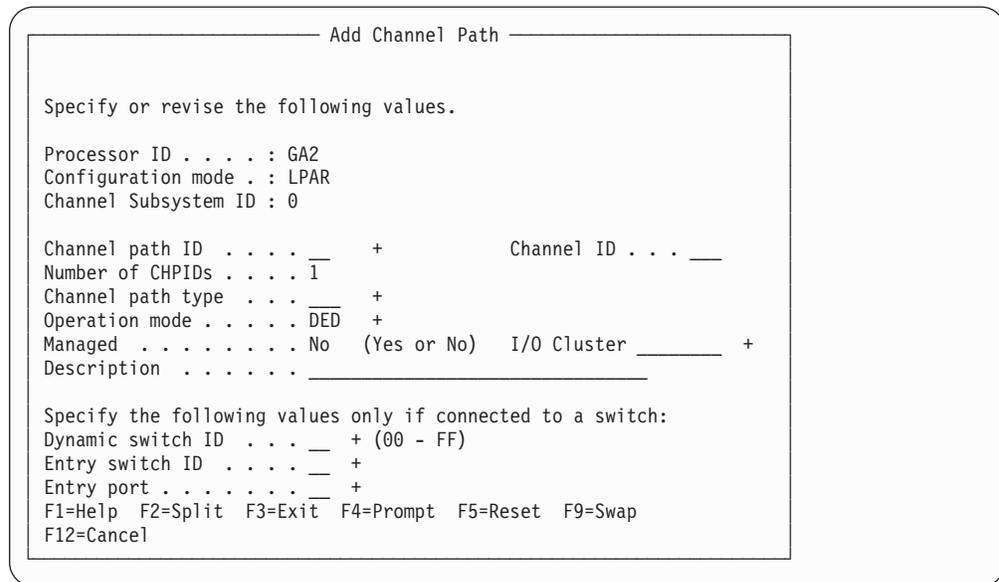


Figure 26. Layout of a data entry dialog

This panel has the following characteristics:

- The position and length of empty entry fields are shown by underscore characters (_) and highlighting. If an entry field contains an entry, the underscore characters are not shown.
- Fields that you cannot change are preceded by a colon (:).
- Fields that you must complete may (depending on the terminal) be highlighted by a different color.
- Input fields with a plus sign (+) indicate that you can prompt information for this field using F4=Prompt. For more information on the prompt facility, refer to “Promptable fields” on page 63.

Working with lists

HCD presents most of the information in form of lists. Depending on the list, there are different methods to select an item from the list or to perform actions. The different types of lists are:

- Numbered selection lists
- Unnumbered single selection lists
- Unnumbered multiple selection lists
- Action lists
- Message lists

This section describes the different types of lists and the respective selection methods.

Numbered selection lists

To select an item from a numbered selection list, type the number you want to select in the input field (precedes the first list item) and press the Enter key. An example of a numbered list is the HCD *Primary Task Selection* panel (see Figure 27), displayed when you start an HCD session.

```

z/OS V2R2 HCD
Command ==> _____

Hardware Configuration

Select one of the following.

- 0. Edit profile options and policies
  1. Define, modify, or view configuration data
  2. Activate or process configuration data
  3. Print or compare configuration data
  4. Create or view graphical configuration report
  5. Migrate configuration data
  6. Maintain I/O definition files
  7. Query supported hardware and installed UIMs
  8. Getting started with this dialog
  9. What's new in this release

For options 1 to 5, specify the name of the IODF to be used.

I/O definition file . . . 'DOCU.IODF00.HCD.WORK'      +

```

Figure 27. *Primary Task Selection* panel

Unnumbered single selection lists

To select a list item from an unnumbered list from which you can select only one item, you can:

- Place the cursor in front of a list item and press the Enter key
- Select a list item with an **S** or a **/** (slash) and press the Enter key.

Figure 28 is an example of a list of IODFs from which you can select one IODF.

You can request a list like the one in Figure 28 by pressing F4=Prompt while your cursor is on the I/O definition file entry field (Figure 27 on page 57). For more details concerning prompting, refer to “Promptable fields” on page 63.

Available IODFs					Row 1 of 24
Command ==> _____					
Select an IODF name.					
IODF Name	MUA	Volume	Size	Created	
DOCU.IODFAA.CTC.SCEN1	Yes	HCDSMS	1080	2008-10-21	
DOCU.IODFAA.CTC.SCEN2A	Yes	HCDSMS	1080	2008-10-21	
DOCU.IODFAA.CTC.SCEN2B	No	HCDSMS	1080	2008-10-21	
DOCU.IODFAA.CTC.SCEN2C	Yes	HCDSM2	1080	2008-10-22	
DOCU.IODF00.HCD.PROD		MIGRAT		2008-02-01	
DOCU.IODF00.HCD.WORK		MIGRAT		2008-02-01	
DOCU.IODF00.HCM.TUTORIAL.PROD	No	HCDSMS	2880	2008-06-17	
DOCU.IODF00.HCM.TUTORIAL.WORK	Yes	HCDSM3	2160	2008-09-12	
DOCU.IODF00.ZOS111.WORK	No	HCDSMS	2880	2008-09-26	
DOCU.IODF01.ZOS110.HCDUG	Yes	HCDSMS	1080	2007-09-27	
DOCU.IODF44.MAINT.WORK		MIGRAT		2008-03-06	
F1=Help	F2=Split	F3=Exit	F7=Backward	F8=Forward	
F9=Swap	F12=Cancel	F14=SortName	F15=SortSize	F16=SortDate	
F22=Command					

Figure 28. Example of a single selection list

Note: For migrated IODFs (with indication MIGRAT in column *Volume*), HCD cannot determine the MUA-status and the size and therefore, columns *MUA* and *Size* remain blank in such cases.

Unnumbered multiple selection lists

To select a list item from an unnumbered list, from which you can select one or more list items, place a slash (**/**) in front of one or several list items and press the Enter key. In some cases, list items in unnumbered multiple selection lists are already preselected. You can overwrite this preselection.

To select a range of objects, you can place a left parenthesis **(** in front of the first item in the range and a right parenthesis **)** in front of the last item in the range. See also “Using the context menu” on page 59.

Figure 29 on page 59 is an example of an unnumbered multiple selection list.

```

Define Access List
Row 1 of 2
Command ==> _____ Scroll ==> CSR
Select one or more partitions for inclusion in the access list.

Channel subsystem ID : 0
Channel path ID .....: 99      Channel pathy type .: FC
Operation mode .....: DED      Number of CHPIDs ...: 1

/CSS ID Partition Name      Number Usage Description
_0 HUGO                      2      OS
_0 LPAR01                    1      CF/OS
***** Bottom of data *****

```

Figure 29. Example of a Multiple Selection List

Action lists

Figure 31 on page 61 is an example of an action list. You see a panel like this one when you want to define, modify, or view I/O devices.

HCD uses the concept of object-to-action processing. That is, to work with an object, you first have to select the object, and then the action. For some actions, you do not have to explicitly select an object; for example, to add an object to an action list.

In an action list, you have several possibilities to perform an action on a list object:

- Using the context menu
- Using the action code
- Typing over data in the panel

Using the context menu

To perform an action on one or several objects in an action list, you can use the context menu:

1. Select an object by placing a slash **/** in front of a list item.
 To select a range of objects, you can place a left parenthesis **(** in front of the first item in the range and a right parenthesis **)** in front of the last item in the range.
2. Press the Enter key. HCD displays a context menu showing all valid actions for the selected objects.

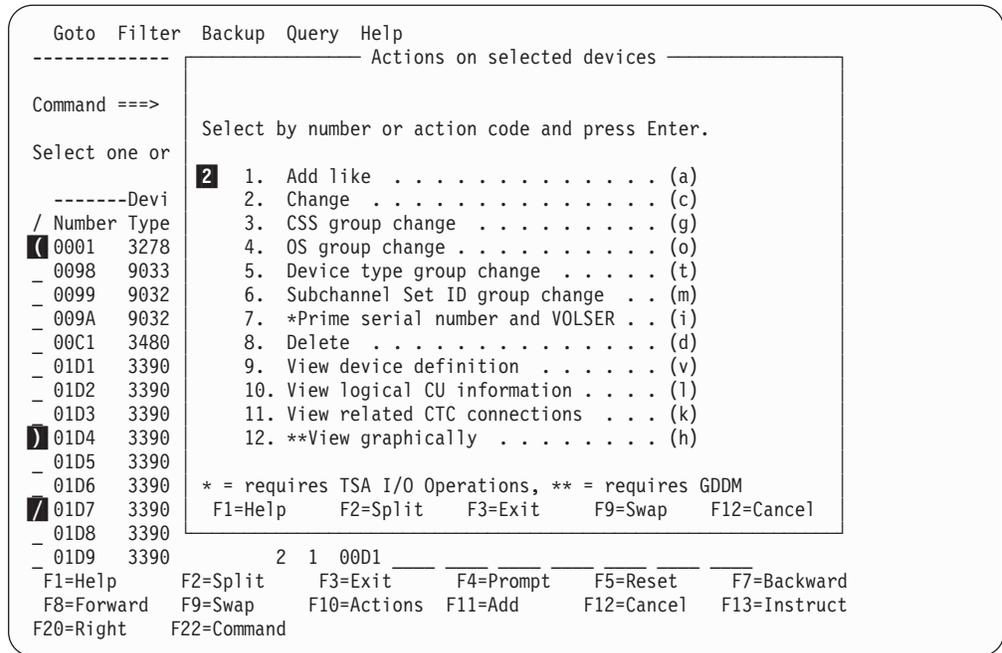


Figure 30. Action list with a context menu

3. Select an action by entering the number or action code (letter in parentheses) in the entry field. After pressing the Enter key, the context menu is closed and the action is performed for the selected rows.

If only one action is possible on an action list panel, the context menu is not shown, but the action is performed immediately after entering a **/** (slash) or an **S**.

In the example in Figure 30, the following objects are changed:

- Objects 0001 through 01D4, and
- Object 01D6

A # marker in the selection column indicates that the row is disabled and not available for processing. For example, coupling facility devices are marked in such a way.

Using the action code

As you get familiar with the dialog, you might find it easier to select a list item and an action in one operation by means of an action code. To do this, enter the action code in the action column, which is the entry field to the left of the associated list item; then press the Enter key. Figure 31 on page 61 shows an example (**d** for delete and **c** for change).

```

Goto Filter Backup Query Help
-----
I/O Device List          Row 1 of 1443 More:  >
Command ==> _____ Scroll ==> PAGE

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

-----Device----- --#--- -----Control Unit Numbers + -----
/ Number  Type +      CSS OS 1--- 2--- 3--- 4--- 5--- 6--- 7--- 8---
- 0001    9032-3      3  4 0001  ___  ___  ___  ___  ___  ___  ___
- 0002    9032-3      3  4 0002  ___  ___  ___  ___  ___  ___  ___
d 0020    BSC1        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
- 0021    BSC2        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
- 0022    BSC2        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
- 0023    BSC1        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
- 0024    BSC1        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
c 0025    BSC2        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
- 0026    BSC1        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
d 0027    BSC1        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
- 0028    TWX         1  4 00E0  ___  ___  ___  ___  ___  ___  ___
- 0029    BSC1        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
- 002A    BSC1        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
- 002B    BSC1        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
- 002C    BSC2        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
- 002D    BSC1        4  4 00E0  ___  ___  ___  ___  ___  ___  ___
# FFFC    CFS         1  4 FFFE  ___  ___  ___  ___  ___  ___  ___

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

```

Figure 31. Action list with action codes

You can select more than one list element and, if necessary, you can specify different action codes for each of the objects. You are first asked to confirm the deletions. Then the other actions (for example, in Figure 31 a change **c**) are performed sequentially row after row.

To get a list of action codes, place the cursor in the action column field and then press the F4=Prompt key. The same context menu is shown when using the **/** in front of a list item and pressing the Enter key. For an example, see Figure 30 on page 60, or Appendix A, “How to navigate through the dialog,” on page 393. If you press the F1=Help key instead of the F4=Prompt key, you get a list from which you can request explanation of the action codes.

Action codes and selection markers: You can also specify individual action codes within a marked range. If you do so, the following rules apply:

- When you press the Enter key, the single action codes are processed. The row selected with selection markers are not processed, they are still shown after processing the action codes.
- When you press the Enter key again, HCD shows the context menu for the rows selected with selection markers.

After successful operation, selection markers and action codes are removed from the list. If the operation is interrupted due to an error, selection markers and action codes not yet processed are still shown. You can remove them by pressing the F5=Reset key.

Typing over existing data

You can also make changes to editable fields by typing in new data or typing over existing data directly on the displayed panel.

You have to press the Enter key to process the changes. HCD then validates the data and displays the panel again. The changes are processed sequentially row after row. Changes, which would need a processing of multiple row changes at once will not be possible.

When you exit action lists, all changes to the list since you last pressed the Enter key are cancelled.

You can use the F5=Reset key to reset the values of all fields that you have changed since you last pressed the Enter key.

Message lists

One single operation can produce multiple messages. In this case, HCD displays a message list. You can then:

- Use the *Explain message* action from the context menu (or action code **e**) to get an explanation of the message.

```

----- Message List -----
Save Query Help
-----
Command ==> _____ Scroll ==> PAGE
Row 1 of 52
Messages are sorted by severity. Select one or more, then press Enter.

/ Sev Msg. ID Message Text
[e] E   CBDG060I A switch and a port must be specified for control unit
#       0090 to connect the control unit to channel path 34 of
#       processor CTC01.
_ E   CBDG060I A switch and a port must be specified for control unit
#       0098 to connect the control unit to channel path 34 of
#       processor CTC01.
_ E   CBDG060I A switch and a port must be specified for control unit
#       0300 to connect the control unit to channel path 61 of
#       processor PR9672.
_ E   CBDG065I The switch control unit 0000 of switch 00 has no channel
#       path attached via the switch.
F1=Help      F2=Split      F3=Exit      F4=Prompt      F5=Reset
F7=Backward  F8=Forward    F9=Swap      F10=Actions    F12=Cancel
F13=Instruct F22=Command

```

Figure 32. Message List

```

----- Message List -----
Command ==> _____ Scroll ==> PAGE
CBDG060I A switch and a port must be specified for control unit
0090 to connect the control unit to channel path 34 of processor
CTC01.
Explanation:
A channel path is connected to a switch, therefore all control
units on this channel path must be connected to a switch also.

System Action:
System waits for user action.

User Response:
Specify a switch and a port for the control unit.
- end -

```

Figure 33. Explanation Message

- Use the *Delete message* action from the context menu (or action code **d**) to delete a message from the message list.
- Select *Save messages* from the *Save* action bar to save the displayed messages in the corresponding message log file.

The messages in Figure 32 on page 62 are sorted by severity, which is indicated in column *Sev*:

- Messages with severity code *T* (terminating) and *E* (error) require your intervention before you can continue to work with HCD.
- For warning messages there are two severity codes *S* (severe warning) and *W* (normal warning). Both severities let you finish the current HCD function. However, you should carefully consider the warnings and it is recommended to remove the reason of the messages.
- Messages preceded by severity code *I* are informational messages.

Promptable fields

The HCD prompt facility reduces what you have to remember, what you have to type, and, possibly, what you have to correct due to typing errors. You can use the prompt facility if there is a plus sign (+) to the right of an entry field (or its column heading on an action list). Just place the cursor on the entry field and press the F4=Prompt key.

HCD then displays a prompt selection menu that lists all the values that are currently valid for the field. For long lists (of I/O devices, for example), HCD first displays a menu to let you limit the values listed (to only DASD devices, for example).

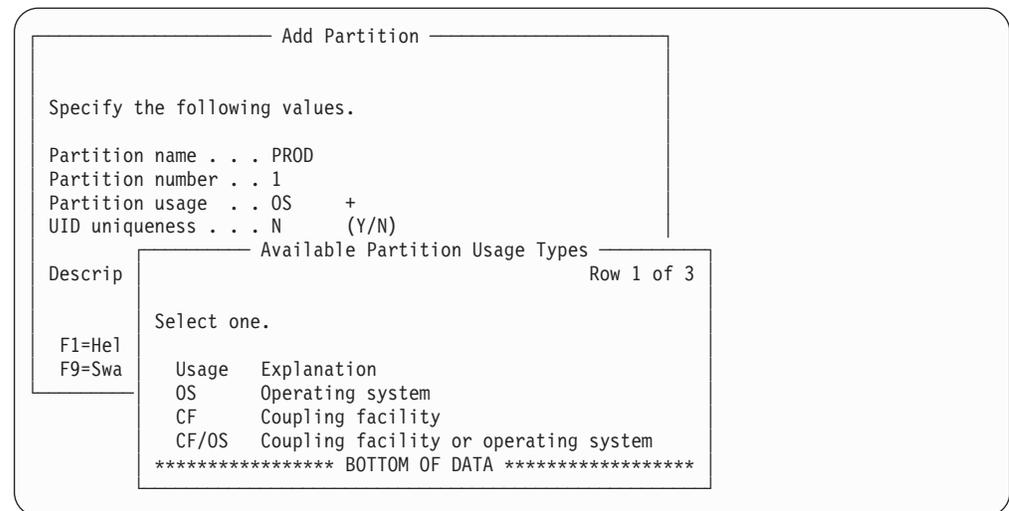


Figure 34. Example of a Prompt Selection Panel

Prompt is also available for the action column. For this purpose you must place the cursor on the action column. HCD displays the same context menu as when using the **/** (slash) in the action column and pressing the Enter key. For an example, see Figure 30 on page 60. Note that there is no '+' sign shown for the action column heading.

To select a value, place the cursor to the left of (or on) that value or select the value with **s** or **/** (slash), then press the Enter key. HCD inserts that value into the entry field or, in case of selecting an action, performs that action immediately.

The prompt list is built dynamically; if a value can be used only once in a configuration then, after it has been selected, either it does not appear in the list again, or it is marked as nonselectable - with the hash sign (#).

Note: It is possible that HCD initially accepts a selection, but rejects it later after further validation when the entered context information is completed.

Some promptable fields can have a large number of choices, for example prompts for device numbers or control unit numbers. In these cases HCD limits the values to a maximum of 128 entries. If the value, you are looking for, is not contained in the choices, you can enter a reference number into the field. If you then press the F4=Prompt key, HCD will display the numbers that follow the reference number.

Commands and function keys

All HCD list panels have a command line on which you can enter the usual ISPF commands, TSO commands, or specific HCD commands.

You can use F22=Command to move the cursor to the command line. When the cursor is in the command line, F22=Command lets you step back through the commands that you have previously entered (including ISPF commands).

The HCD dialog uses 24 function keys, if your ISPF session allows the use of 24 function keys. The function keys assignments can be shown or hidden by entering PFSHOW in the command line.

To perform a function, just press that function key.

Getting help

HCD offers an extensive help facility. From any panel, you can get context-sensitive help by pressing the F1=Help key. Table 5 shortly explains the available types of help and how to obtain it.

Table 5. Online Help Information

Type of Help	Description	How to Get Help
Field	An explanation of what you can enter in a specific entry field on the current panel.	Place the cursor in the entry field and press the F1=Help key.
Extended	Gives information on the content and task of a function panel.	Move the cursor to a non-interactive field in the function panel and press F1, or select Extended help in the action bar, or press F2=Ex_help in field help panels.
Instruction	Specific instructions on what you can do on the current action list panel.	Press the F13=Instruct key. Or select Help from the action bar.
Command	An explanation of the HCD commands that you can enter in the command line.	Place the cursor in the command line and press the F1=Help key. For detailed help on a specific command, type the name of the command and press the F1=Help key.

Table 5. Online Help Information (continued)

Type of Help	Description	How to Get Help
Keys	An explanation of the function keys.	Press the F1=Help key, then press the F9=Keyshelp key from the help panel. Or select <i>Help</i> from the action bar.
Reference Phrase	An explanation of any of the highlighted words or phrases that appear on a help panel.	Place the cursor on the word or phrase and press the Enter key.
Message	An explanation of a message that is displayed on the current panel.	Press the F1=Help key when a message is displayed, regardless of the cursor position.
Help for Help	A general explanation of how to use the help facility.	Press the F1=Help key, then press the F1=Help key again from the help panel. Or select <i>Help</i> from the action bar.
Action Bar	An explanation of the action bar of a panel.	Request extended help, then place the cursor on the reference phrase of the action you want help for, and press Enter.

On a help panel, you can use F5=Window to change the size of the window, to reduce the scrolling required, or to see more of the underlying panel.

Navigating through the dialog

Most tasks you do with HCD are done in one step. However, some tasks need more than one step (*multi-step tasks*). For example, when you define an I/O device, you usually want to define the processor and operating system data for that device at the same time. HCD lets you define this data together, in one task, by displaying a sequence of panels in a predefined order.

Moving forward and backward within a task

The following function keys are available to move forward and backward in a multi-step task.

Enter Displays the next panel in the predefined order. HCD validates your data, but does *not* save it in the IODF until you complete the task. At the end of the task, HCD redisplay the action list that precedes the task.

F3=Exit

Completes the task. HCD saves in the IODF all valid data that you have entered so far in the task (plus any data in the following steps that has been inherited by, for example, an *Add like* action). It then redisplay the action list.

F6=Previous

Available on the second and subsequent steps. It cancels the current step and redisplay the previous panel. Any data that you have entered on the current panel is lost.

F12=Cancel

Cancels the current task. Data entered in the current step is lost but, if you


```

Goto Backup Query Help
-----
                        Operating System Configuration List                Row 1 of 8
Command ==> _____ Scroll ==> CSR

Select one or more operating system configurations, then press Enter. To
add, use F11.

/ Config. ID  Type   Gen  Description                               D/R site OS ID
- DRMVSA     MVS    Yes  generated from MVSA                       _____ DRMVSA
- DRMVSB     MVS    Yes  generated from MVSB                       _____ DRMVSB
- DRMVSC     MVS    Yes  generated from MVSC                       _____ DRMVSC
- DRMVSE     MVS    Yes  generated from MVSE                       _____ DRMVSE
- MVSA       MVS                                _____ DRMVSA
- MVSB       MVS                                _____ DRMVSB
- MVSC       MVS                                _____ DRMVSC
- MVSE       MVS                                _____ DRMVSE
***** Bottom of data *****

F1=Help      F2=Split    F3=Exit     F4=Prompt   F5=Reset    F7=Backward
F8=Forward   F9=Swap     F10=Actions F11=Add     F12=Cancel

```

Figure 36. Operating System Configuration List as example of a Selection List

Using the GOTO command

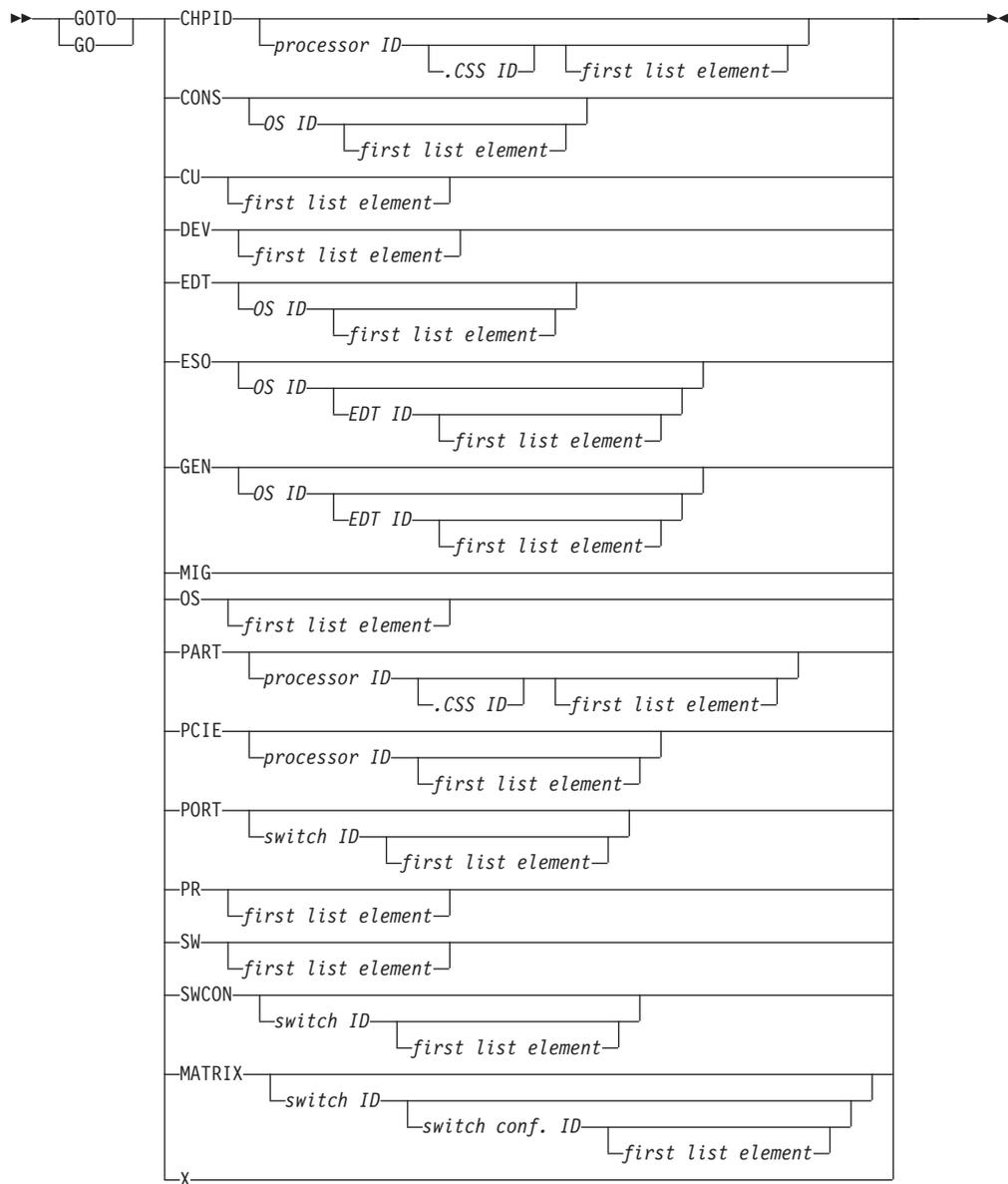
Specify GOTO (or just GO) and the target you want to jump to in the HCD command line. If you type GOTO without a target, a panel is displayed showing the same list of target objects as when using the *Goto* pull-down menu.

You can also jump to list panels that are associated with another object (for example, to the EDT list associated with an operating system). In this case, specify both the target and the object with which it is associated. If you omit this object, a selection list is displayed (for example, the Available Operating System list appears when going to the EDT list).

Optionally, you can also specify the object that will be shown as the first element of a list if you jump to a list panel.

The following figure shows the syntax of the command. The parameters can be separated by either blank, comma, or dot.

GOTO command:



- | | | | |
|--------------|----------------------------|---------------|----------------------|
| CHPID | Channel path list | PART | Partition list |
| CONS | Console list | PCIE | PCIE function list |
| CU | Control unit list | PORT | Port list |
| DEV | Device list | PR | Processor list |
| EDT | EDT list | SW | Switch list |
| ESO | Esoteric list | SWCON | Switch configuration |
| GEN | Generic list | MATRIX | Port matrix |
| MIG | Migrate configuration data | X | Leave HCD |
| OS | OS configuration list | | |

First list element

The object that will be shown as the first element of a list if you jump to a list panel (same as if using the LOCATE command).

Examples: To go to the channel path list of processor SYSA and display CHPID 27 as the first element of the list, enter:

```
GO CHPID SYSA 27
```

To go to the esoterics list of EDT A1 in operating system MVSPROD and display the esoteric group ESO12 as the first element in the list, enter:

```
GO ESO MVSPROD A1 ESO12
```

To specify the processor ID and channel subsystem ID for XMP processors in GOTO PART or GOTO CHPID commands, you use the dot to concatenate both parts:

```
GO PART P2084.3
```

Filtering

If a displayed object list is too long, and scrolling back and forth is getting cumbersome, you may use the *Filter* function that lets you limit the displayed information. The *Filter* function is available when the action bar shows a *Filter* action bar choice. This is the case in a:

- Processor list
- Channel path list
- CTC connection list
- CF channel path connectivity list
- Switch list
- PCIe function list
- Port list
- Control unit list
- Device list
- Esoteric list
- I/O Path list

For example, on the device list, you can limit the displayed information by the following filter criteria:

- Device type
- Device group
- Whether the devices are defined to an operating system or not
- Serial number and description
- Volume serial number
- Device parameters and features (if you navigated to the I/O Device List via the operating system configuration list)
- Whether the devices are connected to a control unit or not
- Up to four control units, to which the devices may be connected
- Subchannel set ID

Rather than seeing all devices defined in the IODF, you see only those devices that are, for example, of a certain type connected to a certain control unit.

How to request filtering is illustrated in the following scenario:

1. Navigate to a processor's *I/O Device List*. Note that for XMP processors you first need to navigate to its *Channel Subsystem List* and then to its *I/O Device List*. Now tab to the action bar and select *Filter* from the action bar. The pull-down menu shown in Figure 37 is displayed:

```

Goto Filter Backup Query Help
-----
Comma 1. Set Filter
       2. Clear Filter
       3. Count rows on (filtered) list
-----
Selec To add, use F11.

Processor ID : TSPROC1 Proc supporting multiple SSs CSS ID : 0

-----Device----- --#-- -----Control Unit Numbers + -----
/ Number  Type +      SS IM OS 1--- 2--- 3--- 4--- 5--- 6--- 7--- 8---
- 0001    3390A      0 0 1 0001 0002 _____
- 0001    3390A      1 2 1 0001 _____
- 0002,4  3390A      1 2 1 0001 _____
- 0002,8  3390A      0 2 1 0001 0002 _____
- 0006,3  3390       2 1 1 0001 _____
- 000A    3380       2 1 1 0001 _____
***** Bottom of data *****

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

```

Figure 37. Filter example

2. Select *Set Filter*. This displays the Filter I/O Device List (Figure 38 on page 71), tailored for the underlying I/O Device List, where you can specify your filter criteria. Note that for example, the 'Subchannel set ID' filter criteria is only available for the I/O Device List if invoked from the Channel Subsystem List of a z9 EC processor or later model, or from the Operating System Configuration List (action code **u** in both cases). Similar panels are displayed for the other lists.

An alternative way to select filtering is to tab to the command line and type in FILTER SET.

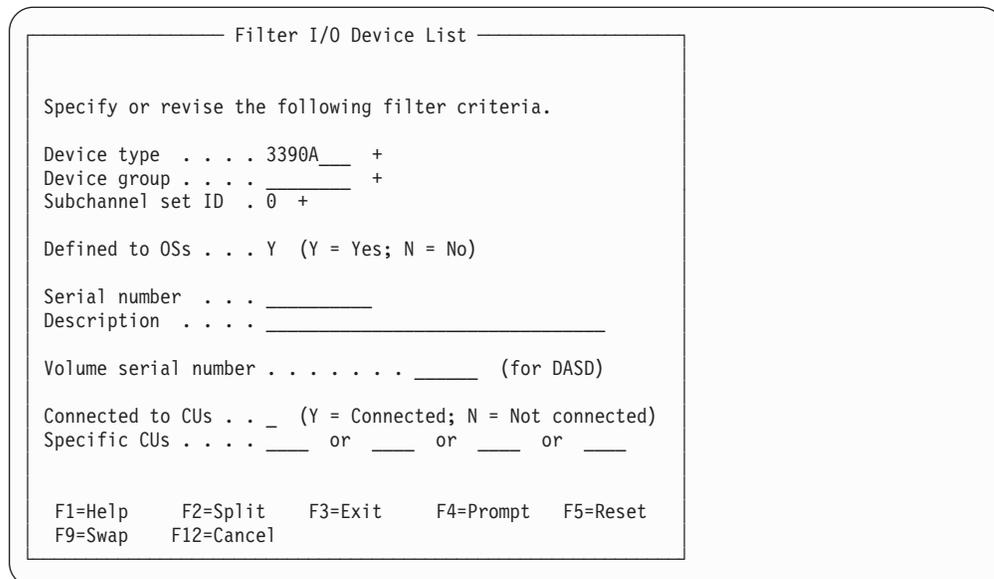


Figure 38. Filter I/O Device List

- On this filter panel you can specify one or more filter criteria. All specified filter criteria must match to display the item. For example, if you specify a device type and a subchannel set where the devices of this type should be defined, and then press the Enter key, the I/O Device List is displayed again, now showing only those devices that match these filter criteria. Figure 39 shows an example of a filtered list. If Filter Mode is displayed in the right top corner of the panel, it indicates that the filter mode is active.

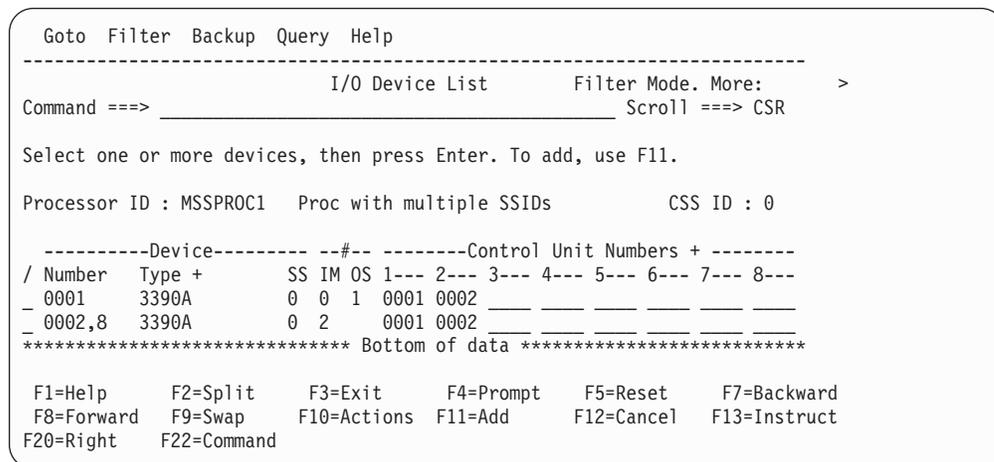


Figure 39. Reduced I/O Device List (using a filter)

Most of the entry fields support wildcards, that means that an asterisk (*) can be specified in front and/or after the specified term. The following table illustrates the wildcard processing using the device type field as an example.

Specified Term	Result
3380	Displays all devices of type 3380
3380*	Displays all devices of type 3380 regardless of its model
338*	Displays all devices whose type starts with '338'
*80	Displays all devices whose type ends with '80'

Specified Term	Result
42	Displays all devices containing the string '42' within the type, for example 3420.

Use Field Help on the Filter panel to get information on whether wildcards are supported or not.

To remove the filter, use *Clear Filter* from the *Filter* action bar, or enter FILTER CLEAR in the command line.

To count the rows on a filtered list, use *Count rows on (filtered) list* from the *Filter* action bar choice, or enter FILTER NUM in the command line. An informational message displays the number of rows in the current list. If this list is filtered, only the rows matching the underlying filter criteria are counted.

Job statement information used in panels

Some of the HCD tasks, invoked from the dialog, generate batch jobs. These batch jobs use EXEC procedures as shown in Table 2 on page 30.

The first time you use a task that generates an HCD batch job, you must specify a JOB control statement for the job. This statement can contain user-specific information, such as an account number and SYSOUT class. For example:

```
//WASR    JOB (3259,BOX10), 'SMITH', NOTIFY=WAS,
//                CLASS=A, REGION=4M, MSGCLASS=F, MSGLEVEL=(1,1)
```

You need to specify the statement only once; it is saved in your user profile and reused for all subsequent HCD batch jobs that you request (in this and following HCD sessions) until you change it. HCD does not validate the JOB statement.

With JCL overwrite statements you can modify the EXEC procedures that are invoked by the job.

Note:

1. A batch job requires a region size large enough to contain the HCD code (4M bytes) as well as the IODF (or two IODFs when you compare two IODFs).
2. A batch job to build an IOCDS must run on the processor on which the IOCDS is to be updated, except for processors configured in a z Systems cluster.
3. A batch job generated by HCD cannot run when the HCD dialog has exclusive access to the same IODF that the batch job uses. You can release the IODF by changing to another IODF or by ending the HCD session.
4. To use the HCD profile options (e.g. UIM_LIBNAME=*) for a batch job that is started from the dialog, you need to allocate the HCD profile data set to xx.HCDPROF, where xx is the job step name.
5. In a sysplex environment, use the JOBPARM parameter to specify in which system of the sysplex you want the batch job to run. For additional information, see Chapter 13, "How to invoke HCD batch utility functions," on page 327.

Table 2 on page 30 lists the HCD tasks that use batch jobs. It also lists their job step names, and EXEC procedure names.

Chapter 6. How to define, modify, or view a configuration

Overview

This information unit describes:

- The possibilities for creating new objects
- The navigation methods through HCD
- A suggested sequence to define a configuration
- Step-by-Step instructions on how to work with (for example, define, change, prime, delete):
 - Operating system configurations
 - EDTs
 - Generics
 - Esoteric groups
 - Processors
 - Channel subsystems (for XMP processors only)
 - Partitions
 - PCIe functions
 - Channel paths
 - Control units
 - Devices
 - Consoles
- The possibilities to view information about objects

The information on how to work with switches is described in Chapter 7, “How to work with switches,” on page 163.

Before using the dialog of HCD to define a hardware configuration, you should have a plan of what your configuration should look like, and what you have to do to accomplish that. Preferably, the requirements of your configuration should be established in a configuration plan. Refer to *z/OS HCD Planning* (for a z/OS, OS/390, or MVS configuration) and *z/VM CP Planning and Administration* (for a z/VM configuration) for a description of what needs to be considered when this plan is prepared.

Creating new objects

You have three possibilities to create new objects: add, add like, and repeat (copy).

Add

Use the F11=Add key to define a new object. Initially, the entry fields contain (where applicable) default values supplied by HCD.

Add like

Use the *Add like* action from the context menu to define a new object that is based on the definition of an existing object. You just have to enter the fields that are different. A field that needs a unique value, such as the object's identifier, is not copied.

Repeat (copy)

The action *Repeat (copy)* from the context menu is similar to *Add like*, but the definitions of all related objects are also copied. For example, if you repeat an operating system configuration, HCD also copies the definitions for all EDTs and consoles, and the connections to all I/O devices attached to that operating system. You can repeat parts of a configuration within the same or to another IODF.

This function is useful when you want to consolidate configuration data from several IODFs into one single IODF or to repeat configuration data (for example, esoterics) that is used several times in an IODF.

When copying parts of a configuration, the source data and the target data are merged.

- If the source object does not exist in the target IODF
The new object is defined using the attributes you specified while copying the object. The objects and connections that are related to the object you want to copy are created with the same attributes as the source objects and connections.
- If the source object already exists in the target IODF
The attributes of the target object are updated according to the attributes of the source object. If related objects or connections do not yet exist, they are created. If they already exist, their attributes are updated according to the attributes of the source.

Navigating through HCD

HCD offers three methods to navigate to objects:

- Centralized navigation
- Hierarchical navigation
- Graphical navigation

Centralized navigation

Centralized navigation means that you always navigate to objects starting from the Define, Modify, or View Configuration Data panel.

The tasks described in this section use this navigation method.

Hierarchical navigation

Hierarchical navigation means that you navigate to objects from top to bottom.

In Figure 40 on page 76 you start, for example, with option 3 to open the Processor List. From the Processor List, you can navigate to channel paths, from channel paths to control units, and finally from the control unit list to the device list. This device list - called Device List (attached) in Figure 40 on page 76- is different from the device list you reach with option 5 from the Define, Modify, or View Configuration panel:

- The devices on this list are limited to the control unit you selected on the preceding control unit list.
- The actions offered on this list differ from the actions available on the device list reached with option 5 from the Define, Modify, or View Configuration panel.

Hierarchical navigation is useful, for example, when you want to use the *Attribute group change* action necessary to change the DYNAMIC parameter for a group of devices. In this case, you have to navigate to the device list via the operating system configuration list.

Graphical navigation

Graphical navigation means that you navigate to objects by viewing a graphical configuration report and jumping to the object lists using the F4=Jump function. Refer to “Create or view graphical configuration reports” on page 257 for information on how to use the graphical configuration report.

The graphical navigation is useful when you prefer a graphical representation of your configuration to navigate from object to object.

Navigation map

Figure 40 on page 76 illustrates how you can navigate from object to object. You can either navigate to objects using the *Work with object* actions from the context menu or using the appropriate action code, for example **s**. Note that you can reach the Generic List with two action codes: **g** (ordered by name) or **p** (ordered by preference value). For information on how to use action codes refer to “Using the action code” on page 60.

Control Unit list and Device list

You reach these lists with option 4 and 5 from the Define, Modify, or View Configuration Data panel. These lists show all devices defined in the IODF.

The Control Unit list (attached) and the Device list (attached) can only be reached from an object higher in the hierarchy. These lists are limited to the object higher in the hierarchy and the actions available on these lists differ from the actions available on the device and control unit lists you reach with option 4 and 5 from the Define, Modify, or View Configuration panel (refer to “Hierarchical navigation” on page 74). You can, for example, limit a control unit list to the control units attached to one specific channel path by opening the list from the channel path list.

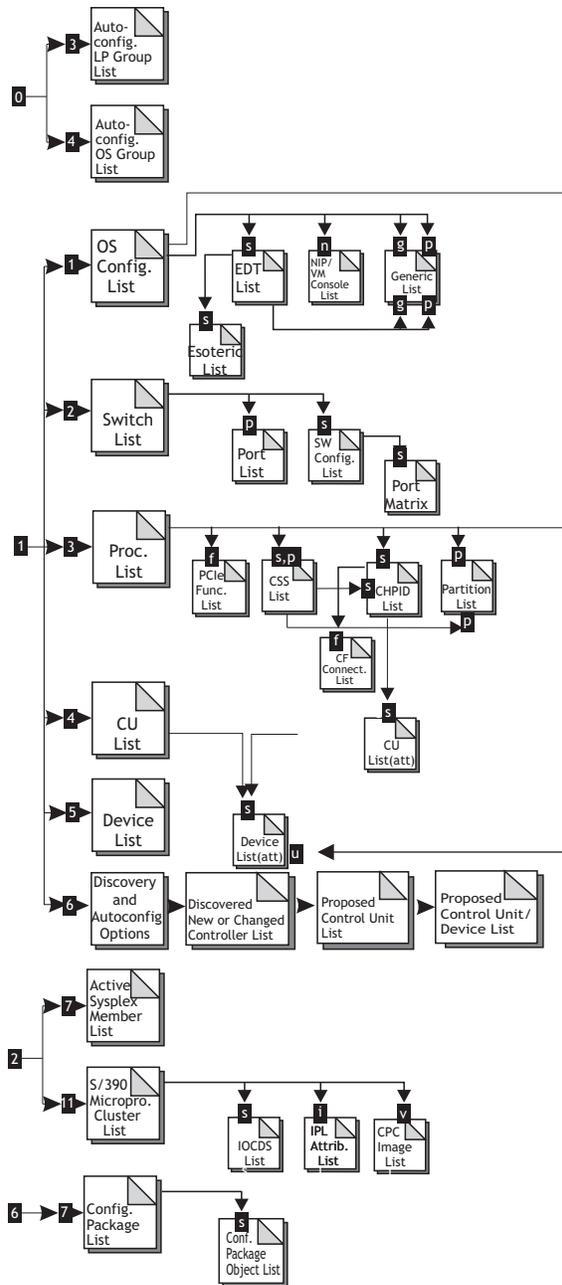


Figure 40. Navigation map

Suggested sequence to define a configuration

A hardware configuration consists of:

- Information needed by z/OS and by z/VM. This is known as the *operating system configuration* or *OS configuration*.
- Information needed by the channel subsystem (CSS). This defines all the hardware resources (such as control units, channel paths, and I/O devices) and how they are connected.

- In some cases, your configuration contains information needed by the switch. For definitions and modifications of switches, refer to Chapter 7, “How to work with switches,” on page 163.

You can define the objects of a configuration in almost any order but at one point you have to connect objects together. You can only connect objects that are already defined; therefore it is useful to define the objects in a logical order. For example, when defining I/O devices during the hardware definition, you are prompted to add devices to existing operating system definitions. Therefore, it is useful to define the operating system before the devices.

The suggested sequence to define a configuration is:

1. Operating systems
2. EDTs (MVS-type only)
3. Esoterics (MVS-type only)
4. Switches (explained in Chapter 7, “How to work with switches,” on page 163)
5. Processors
6. Channel subsystems (for XMP processors)
7. Partitions (if processor in LPAR mode)
8. PCIe functions
9. Channel paths
10. Control units
11. Devices
12. Consoles

Working with operating system configurations

An operating system (OS) configuration defines the data that is used by z/OS or z/VM to build its control blocks. An IODF can contain more than one OS configuration; z/OS is told which one to use at IPL time.

Defining operating system configurations

It is recommended to define the operating system configuration before you define anything else. Proceed as follows to define an operating system configuration:

1. On the *Primary Task Selection* panel, select *Define, modify, or view configuration data* and on the resulting panel the object *Operating system configurations*. HCD displays the Operating System Configuration List of all operating system configurations currently defined in the IODF:

```

Goto Backup Query Help
-----
                        Operating System Configuration List                Row 1 of 8
Command ==> _____ Scroll ==> CSR

Select one or more operating system configurations, then press Enter. To
add, use F11.

/ Config. ID  Type   Gen  Description                               D/R site OS ID
- DRMVSA     MVS    Yes  generated from MVSA                       DRMVSA
- DRMVSB     MVS    Yes  generated from MVSB                       DRMVSB
- DRMVSC     MVS    Yes  generated from MVSC                       DRMVSC
- DRMVSE     MVS    Yes  generated from MVSE                       DRMVSE
- MVSA       MVS    _____
- MVSB       MVS    _____
- MVSC       MVS    _____
- MVSE       MVS    _____
***** Bottom of data *****

```

Figure 41. Operating System Configuration List

If there are no existing configurations in the IODF, the operating system configuration list is empty.

- Use F11=Add to define a new configuration. The data-entry fields are shown in the following figure, with sample data:

```

----- Add Operating System Configuration -----
Specify or revise the following values.

OS configuration ID . . . . . MVSD____
Operating system type . . . . . MVS    +

Description . . . . . z/OS operating system

OS config ID for D/R site . . DRMVSD__ (generated for GDPS)

```

- After you press the Enter key, HCD displays the updated *Operating System Configuration List*.

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 ID 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 I/O definition file* or *Build validated work I/O definition file* tasks, 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 DUPLEX defined storage 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 I/O definition file* or *Build validated work I/O definition file* 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 first or specify profile option UNCOND_GENERATE_DROS=YES. 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 shown in column *Gen* in the *Operating System Configuration List* (Figure 41 on page 78).

Changing operating system configurations

You can change the description of an operating system by just typing over the Description column or by using the *Change* action from the context menu (or action code **c**) on the Operating System Configuration List.

Changing the operating system configuration ID

To change the ID of an operating system, perform the following steps:

1. On the OS Configuration List select the operating system and the *Repeat (copy) OS configurations* action from the context menu (or action code **r**). The Identify Target IODF panel is displayed.
2. Press the Enter key to accept the default target IODF name, that is the IODF you are currently working with. The Repeat Operating System Configuration panel is displayed.

Repeat Operating System Configuration

Specify or revise the following values.

OS configuration ID _____

Operating system type . . . : MVS

Description z/OS operating system

3. Specify the new identifier for the operating system and press the Enter key. HCD displays the OS Configuration List now showing the new operating system.
4. Delete the old operating system by selecting the operating system and the *Delete* action from the context menu (or action code **d**). HCD displays a confirmation panel before showing the updated OS Configuration List.

Repeating (copying) operating system configurations

You can copy operating systems within the same or to another IODF. When copying an operating system, the following related objects and connections are also copied:

- Devices defined for the operating system
- List of consoles (NIPCONs for MVS)
- EDTs including their esoterics and generic groups

In the following example, you copy an operating system to another IODF that already contains an operating system with the same ID that you specify in the repeat panel.

1. Make sure that the operating system in the target IODF has the same operating system type as the one in the source IODF.
2. On the Operating System Configuration List, copy the operating system using the *Repeat (copy) OS configurations* action from the context menu (or action code **r**). The Identify Target IODF panel is displayed.
3. Specify the IODF to which the selected operating system is to be copied. The default IODF is the IODF you are currently working with.
4. On the following Repeat Operating System Configuration panel, specify the required values and press the Enter key.

If a device defined to the operating system already exists in the target IODF, HCD tries to map the device. If more than one device with the same type and number exist, HCD maps the device to the first device found. To avoid this sometimes erroneous mapping, specify a processor and partition on the Repeat Operating System Configuration panel in which the operating system is to be run. Refer to “Migrating additional MVSCP or HCPRIO input data sets” on page 285 for detailed rules when a device is mapped.

Console devices (NIPCONs for MVS) from the source operating system are copied to the beginning of the target's console chain.

5. Because you copy an operating system that already exists in the target IODF, HCD displays a panel to confirm the merging of configuration data.

Deleting operating system configurations

You can delete the complete definition of an operating system using the *Delete* action from the context menu (or action code **d**) on the Operating System Configuration List. This also deletes all EDTs, esoterics, consoles, and connections to devices defined for this operating system.

Working with EDTs

For an MVS-type operating system, you have to define at least one eligible device table (EDT). An EDT can consist of one or more esoteric device groups and names of the generic device types. Esoteric device groups are installation-defined groupings of I/O devices.

An OS configuration can contain more than one EDT; z/OS or OS/390 is told which one to use at IPL time. For background information about I/O device allocation in z/OS that you need to know when defining EDTs and esoteric groups, refer to *z/OS HCD Planning*.

Defining EDTs

Before you can define EDTs, you must have defined an operating system. You define an EDT as follows:

1. On the *Primary Task Selection* panel, select *Define, modify, or view configuration data* and on the resulting panel the object *Operating system configurations*. HCD displays the Operating System Configuration List of all operating system configurations currently defined in the IODF.
2. On the Operating System Configuration List, select the OS configuration and the *Work with EDTs* action from the context menu (or action code **s**). HCD displays the EDT List.

```

----- EDT List -----
Goto Backup Query Help
-----
Command ==> _____ Scroll ==> PAGE
Row 1 of 1
Select one or more EDTs, then press Enter. To add, use F11.
Configuration ID . : OPSYS01      MVS or z/OS operating system
/ EDT Last Update By      Description
_ AS 1994-10-04 DOCU      basic
***** BOTTOM OF DATA *****

```

If there are no EDTs defined in the IODF, the EDT list is empty.

- Use F11=Add to add a new EDT. The data-entry fields are shown in the following figure, with sample data:

```

----- Add EDT -----
Specify the following values.
Configuration ID . : OPSYS01      MVS or z/OS operating system
EDT identifier . . . A1
Description . . . . special

```

- After you press the Enter key HCD displays the updated EDT List.

Changing EDTs

You can change the description of an EDT by just typing over the Description column or using the *Change* action from the context menu (or action code **c**) on the EDT List.

Changing the EDT ID

To change the ID of an EDT, perform the following steps:

- On the EDT List select the EDT and the *Repeat (copy) EDTs* action from the context menu (or action code **r**). The Identify Target IODF panel is displayed.
- Press the Enter key to accept the default target IODF name, that is the IODF you are currently working with. The Repeat EDT panel is displayed.

```

----- Repeat EDT -----
Specify or revise the following values.
Configuration ID . . OPSYS01  +
EDT identifier . . . _
Description . . . . special

```

- Specify the new identifier for the EDT and press the Enter key. HCD displays the EDT List now showing the new EDT.
- Delete the old EDT by selecting the EDT and the *Delete* action from the context menu (or action code **d**). HCD displays a confirmation panel before showing the updated EDT List.

Repeating (copying) EDTs

You can copy EDTs within the same or to another IODF. When copying an EDT, the esoteric groups and the VIO eligible parameter are also copied. Perform the following steps to repeat an EDT:

1. Make sure that the devices are already defined to the target operating system.
2. On the EDT List, select an EDT and the *Repeat (copy) EDTs* action from the context menu (or action code **r**). The Identify Target IODF panel is displayed.
3. Specify the IODF to which the selected operating system is to be copied. The default IODF is the IODF you are currently working with.
4. On the following Repeat EDT panel, specify the required values and press the Enter key.

If the EDT already exists in the target IODF, the esoteric groups and their devices are merged. In this case, HCD displays a panel to confirm the merging of data.

Deleting EDTs

You can delete the definition of an EDT using the *Delete* action from the context menu (or action code **d**) on the EDT List. This also deletes the esoterics.

Working with generics

Device types with similar characteristics are logically grouped together and assigned a name to by the system. Such a group is called a generic device type. Reference to a generic device type is made by its name. To request a device allocation, a user can specify a generic device type rather than a specific device number. z/OS or OS/390 then allocates a device from the specified generic device type.

1. On the *Primary Task Selection* panel, select *Define, modify, or view configuration data* and on the resulting panel the object *Operating system configurations*. HCD displays the OS Configuration List of all operating system configurations currently defined in the IODF.
2. On the Operating System Configuration List, select the OS configuration and the *Work with EDTs* action from the context menu (or action code **s**). HCD displays the EDT List.
3. To change the generics, select either the action *Work with generics by name* from the context menu (or action code **g**), or the action *Work with generics by preference value* (or action code **p**) on the EDT List. On the resulting panels you can then change the VIO indicator, the preference value for a generic, and you can display a subsequent panel that lists the devices belonging to the specific generic.

Working with esoteric groups

An esoteric device group identifies the I/O devices that are included in that group. The name you assign to an esoteric device group is called the esoteric name. To request allocation of a device from an esoteric device group, specify the esoteric name on the UNIT parameter of a JCL DD statement. The name *esoteric device group* is often shortened to *esoteric group* or simply *esoteric*.

Defining esoteric groups

You can define which esoteric device groups are in each EDT after you have defined the OS configuration. But you cannot assign I/O devices to an esoteric device group until the devices have been defined.

1. On the *Primary Task Selection* panel, select *Define, modify, or view configuration data* and on the resulting panel the object *Operating system configurations*. HCD displays the OS Configuration List of all operating system configurations currently defined in the IODF.
2. On the Operating System Configuration List, select the OS configuration and the *Work with EDTs* action from the context menu (or action code **S**). HCD displays the EDT List.
3. On the EDT List, select the EDT and the *Work with esoterics* action from the context menu (or action code **S**). HCD displays the Esoteric List.

```

Goto Filter Backup Query Help
-----
Row 1 of 1

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

Configuration ID . : OPSYS01      MVS or z/OS operating system
EDT identifier . . : AS          basic

/ Esoteric VIO  Token State
_ ES001    No   _____ No device defined
***** BOTTOM OF DATA *****

```

4. Use F11=Add to add a new esoteric group. The data-entry fields are shown subsequently, with sample data:

```

----- Add Esoteric -----

Specify the following values.

Esoteric name . . . ES002__
VIO eligible . . . No   (Yes or No)
Token . . . . . _____

```

5. After you press the Enter key HCD displays the updated Esoteric List.

Assigning devices to esoterics

You must define the I/O devices before you can assign them to an esoteric group. This is described in “Defining devices” on page 134. The State column on the Esoteric List indicates the esoteric groups that have no devices defined; so you can check later that your groups are properly defined.

If I/O devices are already defined, you can assign them to esoteric groups as described in “Adding devices to esoterics” on page 84.

Changing esoteric groups

You can change the following data of an esoteric group by just typing over the corresponding columns or by using the *Change* action from the context menu (or action code **C**) on the Esoteric List:

- Esoteric name
- VIO eligible parameter

- Token

Adding devices to esoterics

You have two possibilities to add devices to esoterics:

While defining devices

When you define a device, HCD automatically prompts you to define the device to an operating system, then to an EDT and esoterics. Refer to “Defining devices” on page 134 for a step-by-step instruction on how to do that.

While modifying esoterics

You can add existing devices to esoterics at any time from the Esoteric List as follows:

1. On the Esoteric List, select the esoteric and the *Assign devices* action from the context menu (or action code **s**). HCD displays the Assign/Unassign Devices to Esoterics panel.
2. On the Assign/Unassign Devices to Esoterics panel, overwrite the values in the Assigned column to assign (YES) or unassign (NO) devices to the esoterics of the selected row.

If you do not want to assign a complete group of devices, you can limit the range by specifying a starting number and the number of devices. If you omit the number of devices, 1 is assumed.

Assign/Unassign Devices to Esoteric

Goto Filter Backup Query Help

Row 1 of 4

Specify Yes to assign or No to unassign.

Configuration ID . : OPSYS01 MVS or z/OS operating system
 EDT.Esoteric . . . : A1.ES001 VIO eligible . : No

Devices	Device Type	Generic Name	Assigned	Starting Number	Number of Devices
0001,1	3278-3	3277-2	Yes	---	---
0098,1	9033	SWCH	No	---	---
00C1,1	3480	3480	No	---	---
01D1,8	3390	3390	No	---	---

3. Press the Enter key to process the changes. Then press the F3=Exit key to return to the Esoteric List.

Repeating (copying) esoteric groups

You can copy esoterics within the same or to another IODF. When copying an esoteric, the list of assigned devices is also copied.

Perform the following steps to repeat esoterics:

1. Make sure that the devices are already defined to the target operating system.
2. On the Esoteric List select an esoteric and the *Repeat (copy) esoterics* action from the context menu (or action code **r**). The Identify Target IODF panel is displayed.
3. Specify the IODF to which the selected operating system is to be copied. The default IODF is the IODF you are currently working with.
4. On the following Repeat Esoteric panel, specify the required values and press the Enter key.

If the esoteric already exists in the target IODF, the devices defined for the esoteric are merged. In this case, HCD displays a panel to confirm the merging.

Deleting esoteric groups

You can delete the definition of an esoteric using the *Delete* action from the context menu (or action code **d**) on the Esoteric List.

Working with processors

A note on terminology:

Throughout this document, the following terms are used:

XMP processors and SMP processors

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

For XMP processors, the HCD dialog offers methods to explicitly define multiple logical channel subsystems (MCSS). For SMP processors, the single channel subsystem is implicitly defined with the processor.

You can define more than one processor in an IODF and for each defined processor you can configure processor-related data for further use by the CSS.

For processors that are physically partitioned, you must define each physical partition as an individual processor.

Defining processors

Define a processor as follows:

1. On the *Primary Task Selection* panel, select *Define, modify, or view configuration data* and on the resulting panel the object *Processor*. HCD displays the Processor List of all processors currently defined in the IODF.

```
Goto Filter Backup Query Help
-----
Processor List          Row 1 of 3 More:  >
Command ==> _____ Scroll ==> CSR

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

/ Proc. ID Type + Model + Mode+ Serial-# + Description
- PROC1  2817  M49   LPAR  _____ z196 first processor
- PROC2  2817  M32   LPAR  _____ z196 second processor
- PROC3  2818  M10   LPAR  _____ z114 BC
- PR2827 2827   H75   LPAR  _____ zEC12 with PCIe functions
***** Bottom of data *****
```

2. Use F11=Add to add a new processor. The data-entry fields are shown in the following figure, with sample data:

```

      Add Processor
-----
Specify or revise the following values.

Processor ID . . . . . PROC4
Processor type . . . . . 2817 +
Processor model . . . . . M80 +
Configuration mode . . . . . LPAR +
Number of channel subsystems . . 4 +

Serial number . . . . . 1234562817
Description . . . . . IBM z196 processor

Specify SNA address only if part of a System z cluster

Network name . . . . . _____ +
CPC name . . . . . _____ +

Local system name . . . . . _____

```

Figure 42. Add a processor

On the **Add Processor** panel, you can specify the network name and the CPC name, when the processor is configured in a z Systems cluster. If you specify a SNA address, refer to “Security-related considerations” on page 359 for specific access authority.

A local system name (eight alphanumeric characters) is used by the CPC to identify itself when establishing a coupling facility connection using channel paths of types such as CIB or CS5. You need to specify local system names for both the target and source processors of the coupling facility. If you do not enter a local system name, and a CPC name is given, the local system name defaults to the CPC name.

For SMP processor you must not specify a local system name.

Use Prompt on the Add Processor panel for the SNA addresses for those CPCs that are currently configured in the z Systems cluster.

- Depending on the processor type/model, there may be more than one support level for the processor type. The support level defines the supported channel path types, and the features. If the processor has several support levels, HCD displays another panel showing a list of available support levels for the processor (in our example, for a processor of type 2964).

```

      Available Support Levels
-----
Row 1 of 3 More: >
Command ==> _____

Select the processor support level which provides the processor
capabilities you want to use.

Support Level
2964 support, ISM, RCE, UID
2964 support
2964 support, ISM, RCE
***** BOTTOM OF DATA *****

F1=Help      F2=Split    F3=Exit     F7=Backward  F8=Forward
F9=Swap      F12=Cancel  F20=Right   F22=Command

```

Figure 43. Available support levels

Select the appropriate support level. HCD uses this level when validating the configuration for this processor. It relates to the installed microcode.

Note: On the Available Support Levels panel you can retrieve an explanation of the processor support level for z Systems processors. Position the cursor on the support level description and press the F1 key to get an enumeration of functions provided by this support level.

4. After you press the Enter key HCD displays the updated Processor List. You can now use the F20=Right key to scroll to the right to see the SNA address, if you have defined one.

Changing processors

You can change the following data of the processor:

- Type
 - Model
 - Configuration Mode
 - Serial Number
 - Description
 - Network name
 - CPC name
 - Local system name
1. On the *Processor List*, select the processor and apply the *Change* action from the context menu (or action code **c**). The *Change Processor Definition* dialog is displayed.

Change Processor Definition

Specify or revise the following values.

Processor ID : PROC4

Support level:
XMP, 2817 support, SS 2, 32 CIB CF LINKS

Processor type 2817 +

Processor model M80 +

Configuration mode LPAR +

Serial number 1234562817 +

Description IBM z196 processor

Specify SNA address only if part of a System z cluster:

Network name _____ +

CPC name _____ +

Local system name _____

2. To change the processor type, or model, overwrite the old processor type, or model values, and press the Enter key. To change the support level for the same processor type, move the cursor to the Support level line and press the Enter key. If you have installed a new processor type or model, or a new processor support level, you need to upgrade the processor definition within HCD. HCD selects the proper configuration rules that are dependent on the processor type, and support level, and generates the correct input for the IOCDS download process.

Note: If the processor change leads to error message CBDA102I, make sure that the new processor type/model supports the same configuration rules as the old processor type, for example, that the same channel path types are supported. Subsequent messages indicate an invalid support level. You first have to update your configuration according to the new processor type/model before you can change the processor.

Note: In case of different configuration rules HCD might provide defaults or clear conflicting values. Appropriate messages will be displayed.

3. If more than one support level is available for the processor type, the Available Support Levels panel is displayed where you can select the correct support level for your processor.

Note: On the Available Support Levels panel you can retrieve an explanation of the processor support level for z Systems processors: Position the cursor on the processor support level description and press the F1 key to get an enumeration of functions provided by this support level.

4. The Update Channel Path Identifiers panel is displayed. This panel shows the old channel path definitions, which you can change according to your new processor configuration.

Update Channel Path Identifiers

Row 1 of 24

Command ==> _____ Scroll ==> CSR

Specify any changes to the channel path identifiers in the list below.

Processor ID : PROC1
 Channel Subsystem ID . . : 0

CHPID	Type	Side	Until	CHPID	New CHPID +
24	OSD		—		24
25	OSC		—		25
26	OSM		—		26
27	CFP		—		27
28	CFP		—		28
29	CNC		—		29
2A	FC		—		2A
2B	FC		—		2B
2C	FCP		—		2C

The column New CHPID shows the channel path IDs to which the values of column CHPID are mapped by default.

- You can overtype the values of the column New CHPID. Leave the column Until CHPID blank. Press the Enter key to move the old channel path IDs to the new channel path IDs.
 - If you enter a value in the column Until CHPID, you move all defined CHPIDs in the range defined by the channel paths specified in the CHPID and Until CHPID column to the new range that starts with the value in the New CHPID column. If the values for the CHPID column have a 'gap', the 'gap' is also reflected in the New CHPID column range. When you entered a value in the Until CHPID column, press the Enter key. The Update Channel Path Identifiers panel is redisplayed where the new range is resolved in the New CHPID. The Until CHPID column is shown as blank. To process the updates, press the Enter key again.
5. The source and target CHPIDs of a CIB coupling connection are each given the local system name of the processor to which they will connect. This is automatically done by HCD at the moment when the IOCP input file is built. Therefore, you should be aware of the consequences of changing the local

system name. Consider, for example, a scenario, where during a processor upgrade, you want to adapt an existing local system name to match with a new CPC name. Then the I/O configuration of the target processor of an existing CIB connection also changes and requires a dynamic activate or a POR (in case of a stand-alone CF CPC) to re-establish the CF links. Therefore, HCD issues the CBDG400I warning message whenever a user changes a processor's local system name.

Note: If the processor upgrade changed the SYSTEM value of the corresponding IOCP configuration, it may not be possible to do an IOCDS download. It is only possible to perform an IOCDS download for specific processor types because IOCP validates the generated IOCP statements according to the configuration rules of the executing processor. See “Supported hardware report” on page 425 for the processor type which allows you to download an IOCDS in preparation for a processor upgrade.

If a processor type change leads to change of the SYSTEM value of the corresponding IOCP configuration, see “Build an IOCDS” on page 212 and “Build z Systems cluster IOCDSs” on page 215.

Changing the processor ID

To change the ID of a processor, perform the following steps:

1. On the Processor List, select the processor and the *Repeat (copy) processor configurations* action from the context menu (or action code **r**). The Identify Target IODF panel is displayed.
2. Press the Enter key to accept the default target IODF name, that is the IODF you are currently working with. The Repeat Processor panel is displayed.

Repeat Processor

Specify or revise the following values.

Processor ID PROC5__

Processor type : 2817
 Processor model : M80
 Configuration mode : LPAR

Serial number 1234562817 +
 Description IBM z196 processor

Specify SNA address only if part of a z Systems cluster:

Network name _____ +
 CPC name _____ +

3. Specify the new identifier for the processor and press the Enter key. HCD creates a new processor with the same characteristics and connections as the old one.
4. On the resulting Processor List, delete the old processor by selecting the processor and selecting the *Delete* action from the context menu (or action code **d**). HCD displays a confirmation panel before showing the updated Processor List.

Repeating (copying) processors

You can copy processors within the same or to another IODF. When copying a processor, the following related objects and connections are also copied:

- Channel subsystems (for XMP processors only)
- Partitions
- Channel paths
- PCIe functions
- Control units
- Devices

When you copy a processor within the same IODF, you create a new processor. When you copy a processor into a different IODF, you can either create a new processor with the *Repeat (copy) processor configuration* action, or, with the same action, merge the configuration of the source processor into an existing configuration of the target processor.

Before copying the processor to an existing processor in another IODF (merge the processor configurations), check the following:

1. Make sure that source and target processor have the same type-model and support level.
2. Check the partition usage type if the processor has several partitions.
You cannot replace a partition that has a different usage type (except if there are no channel paths defined for this partition):

Usage Type in Source IODF	Matching	Usage Type in Target IODF
OS	→	OS
CF	→	CF
CF/OS	→	CF, OS, or CF/OS

If the partition usage types do not match, change them in the target IODF.

3. If a channel path already exists in the target IODF, make sure that it has the same type.
4. If the target channel path connects to another dynamic switch than the source channel path, disconnect the channel in the target IODF.
5. To replace a shared channel path mode by a dedicated or reconfigurable channel path mode, change the mode or delete the incompatible channel path in the target IODF.
6. If a processor is copied that has a local system name defined, but the CPC name is changed for the copied processor, the local system name is left unchanged. Whenever a local system name has been defined, either explicitly or by default, a change only happens by explicit user action.

Copy a processor as follows:

1. Copy the processor using the *Repeat (copy) processor configurations* action from the context menu (or action code **r**) on the Processor List. The Identify Target IODF panel is displayed.
2. Specify the IODF to which the selected processor configuration is to be copied. The default IODF is the IODF you are currently working with.
3. On the following Repeat Processor panel, specify the required values and press the Enter key.

Definitions for source partitions are merged with the definitions of the target partitions. If a target partition has another partition number than the source partition, HCD keeps the target partition number.

If control units, devices, CHPIDs, or PCIe functions already exist in the target IODF (same number and type), HCD tries to map them. "Migrating additional

IOCP input data sets” on page 283 explains in detail when a device or control unit is mapped. The general rule is, that target definitions are updated from the source definitions.

See “Explicit device candidate lists” on page 101 on how to handle explicit device candidate lists.

Copying an SMP processor to an XMP channel subsystem

You can copy the configuration of an SMP processor into an XMP channel subsystem within the same or to another IODF. When copying an SMP processor to a CSS, the following related objects and connections are also copied:

- Partitions
- Channel paths
- Control units
- Devices

You can use this function for the following alternative tasks:

1. **alternative 1:** merge the source processor with an existing CSS in the target processor
2. **alternative 2:** copy the source processor to a new CSS in the target processor

Copy an SMP processor to a channel subsystem as follows:

- On the Processor List, select an SMP processor and the action *Copy to channel subsystem... (SMP)* from the context menu (or action code **y**).
- Specify the IODF to which the selected operating system is to be copied. The default IODF is the IODF you are currently working with.
- On the following panel, specify your target processor ID and channel subsystem ID. Make sure that the target processor supports multiple logical channel subsystems.

Copy to Channel Subsystem

Specify or revise the following values.

Source processor:
Processor ID : PROCA0 Processor with single CSS

Target channel subsystem:
Processor ID : PROC4__ +
Channel subsystem ID . . 0 +

After pressing the Enter key:

- **for alternative 1**, you are prompted to confirm or cancel the merging of the source processor into the existing target CSS. A message will inform you about the success of the operation.
- **for alternative 2**, on the Add Channel Subsystem panel, you are prompted to define the ID for the new CSS and the maximum number of allowed devices.

```

----- Add Channel Subsystem -----
Specify or revise the following values.

Processor ID . . . . . : P2964          performance test system

Channel Subsystem ID . . . 4 +
Description . . . . . CSS 4 of processor P2964_____

Maximum number of devices
in subchannel set 0 . . 65280 +
in subchannel set 1 . . 65535 +
in subchannel set 2 . . 65535 +
in subchannel set 3 . . 65535 +

```

If, by chance, partition names from the source processor already exist in the target processor, you are prompted to specify new names for those partitions.

Priming processor data

You can prime your I/O configuration in a work IODF with the processor serial number for the active processor.

To prime, select the action *Prime serial number* from the context menu (or action code **i**) on the Processor List.

The Confirm Priming Processor List shows the selected processors with the sensed data for the processor type and serial number of the active processor and their corresponding definitions in the IODF. If the processor type of the active processor and the defined processor match, they are shown in the Confirm Priming Processor List.

The sensed data for the processor serial numbers can be confirmed before being taken into the IODF. If a value is blanked out, the defined IODF value is not changed. If you use the F12=Cancel key, none of the sensed values is taken.

```

----- Confirm Priming Processor List -----
Row 1 of 1
Command ==> _____ Scroll ==> CSR

Press Enter to confirm priming, or Cancel to leave the list.
A blank value will not change the IODF definition.

Processor ----- Type ----- --- Serial Number ---
ID      actual   defined   sensed   defined
R35     2817-M49    2817-M49    089F252817
***** Bottom of data *****

F1=Help    F2=Split    F3=Exit    F5=Reset    F7=Backward
F8=Forward F9=Swap     F12=Cancel

```

Deleting processors

You can delete the definition of a processor using the *Delete* action from the context menu (or action code **d**) on the Processor List. If you delete a processor, all channel paths, partitions, and connections to control units and devices for that processor are also deleted; the control units and devices are not deleted.

Working with channel subsystems

Note:

All tasks described in this unit's subsections are only available for XMP processors.

For XMP processors, you may define multiple logical channel subsystems (up to n), identified by a unique 1-digit hexadecimal number (range 0 through n-1). For each channel subsystem you may define its own set of partitions and CHPIDs.

Defining channel subsystems

Define a channel subsystem as follows:

- On the Processor List, select an XMP processor and the action *Work with channel subsystems . . (XMP)* from the context menu (or action code **s**). HCD displays the Channel Subsystem List.

```
Goto Backup Query Help
-----
Channel Subsystem List      Row 1 of 6 More:  >
Command ==> _____ Scroll ==> PAGE

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

Processor ID . . . : P2964      performance test system

  CSS Devices in SS0   Devices in SS1   Devices in SS2   Devices in SS3
/ ID Maximum + Actual Maximum + Actual Maximum + Actual Maximum + Actual
- 0  65280   16825   65535    0    65535    0    65535    0
- 1  65280   16709   65535    0    65535    0    65535    0
- 2  65280   14587   65535    0    65535    0    65535    0
- 3  65280   15239   65535    0    65535    0    65535    0
***** Bottom of data *****
```

- Use F11=Add to define a new channel subsystem. The data-entry fields are shown in the following figure, with sample data:

```
Add Channel Subsystem

Specify or revise the following values.

Processor ID . . . . . : P2964      performance test system

Channel Subsystem ID . . . 4 +
Description . . . . . CSS 4 of processor P2964 _____

Maximum number of devices
in subchannel set 0 . . 65280 +
in subchannel set 1 . . 65535 +
in subchannel set 2 . . 65535 +
in subchannel set 3 . . 65535 +
```

Define the ID for the new CSS and the maximum number of allowed devices. For z9 EC processors or later, you can specify the maximum number of devices for more than one subchannel set.

Pressing the Enter key brings you back to the Channel Subsystem List.

Starting with IBM System z10 processors, the maximum number of channel subsystems is predefined during creation of this processor.

Repeating (copying) channel subsystems

You can copy channel subsystems within the same or to another IODF:

1. merge the source CSS with an existing target CSS in the same processor
2. copy a source CSS to a new target CSS in the same processor
3. copy a source CSS to another XMP processor

Copy or repeat a channel subsystem as follows:

- On the Processor List, select an XMP processor and the action *Work with channel subsystems . . (XMP)* from the context menu (or action code **s**).
- On the Channel Subsystem List, for the source CSS, select action *Repeat (Copy) channel subsystem* from the context menu (or action code **r**).
- Specify the IODF to which the selected operating system is to be copied. The default IODF is the IODF you are currently working with.
- On the following panel, specify your target processor and CSS ID:

```
Repeat Channel Subsystem

Specify or revise the following values.

Processor ID . . . . . PROC4      +
Channel subsystem ID . . 2      +
```

- Depending on what you want to do, continue as follows:
 - **To merge source and target CSS**, you are prompted for confirmation for merging the source CSS data into the target CSS data. If you confirm, you are prompted to specify new partition names, because the partitions within an XMP processor must be unique:

```
Specify New Partition Names                               Row 1 of 2

Command ==> _____

The partitions listed have already been defined in the
target processor. Specify new names and press ENTER.

Partition  New Name
LPAR01     _____
LPAR02     _____
***** Bottom of data *****
```

A message indicates the successful processing.

- **To copy the source CSS into a new target CSS**, you can specify or revise certain characteristics of the target CSS:

```

----- Add Channel Subsystem -----

Specify or revise the following values.

Processor ID . . . . . : P2964           performance test system

Channel Subsystem ID . . . 4 +
Description . . . . . : CSS 4 of processor P2964_____

Maximum number of devices
  in subchannel set 0 . . 65280 +
  in subchannel set 1 . . 65535 +
  in subchannel set 2 . . 65535 +
  in subchannel set 3 . . 65535 +

```

You are prompted to specify new partition names. Also, a message indicates the successful processing.

- **To copy a source CSS to another XMP processor**, the processing is identical as copying into a new target CSS. However, specifying new partition names is only necessary if the partition names of the source processor already exist in the target processor.

Note: CHID values are not copied.

Copying/repeating channel subsystems with CTC connections

When copying channel subsystems or partitions within the same IODF, valid CTC connections in the source are not automatically copied and changed in the target configuration. Instead, HCD displays the *CTC Connection Update List*. This list shows all valid CTC connections of the source configuration that need an update before being copied to the target. From this list, you can select those connections that you want to copy. For each selected CTC connection, HCD updates the CUADD definition such that the valid CTC connection is moved from the source to the target. For FCTC control units, HCD generates the full-byte CUADD value for target XMP processors, consisting of the concatenation of CSS and MIFID.

```

                                CTC Connection Update List                                Row 1 of 8
Command ==> _____ Scroll ==> PAGE

Target processor ID: P2084
Target CSS ID . . . : 2

Select CTC connections to be moved to target configuration. Then,
press Enter.

-----CTC or FC side----- -----CNC/FCV or FC side-----
/ Proc.CSSID Part.   Devices CH CU Proc.CSSID Part.   Devices CH CU
- TREX.0 TCSS0LP2 8004,2 20 8004 P2064M2 FREE2LP2 8000,2 21 8000
- TREX.0 TCSS0LP2 9000,2 20 9000 RAPTOR RAPOS4 9004,2 10 9004
- TREX.0 TCSS0LP2 9002,2 20 9002 RAPTOR RAPMIX9 9004,2 10 9004
- TREX.0 TCSS0LP3 8006,2 20 8006 P2064M2 FREE2LP5 8002,2 21 8002
- TREX.0 TCSS0LP3 9000,2 20 9000 RAPTOR RAPOS4 9006,2 10 9006
- TREX.0 TCSS0LP3 9002,2 20 9002 RAPTOR RAPMIX9 9006,2 10 9006
- TREX.0 TCSS2LP2 8000,2 21 8000 P2064M0 FREE0LP2 8004,2 20 8004
- TREX.0 TCSS2LP5 8002,2 21 8002 P2064M0 FREE0LP3 8006,2 20 8006
*****
***** BOTTOM OF DATA *****
F1=Help F2=Split F3=Exit F5=Reset F7=Backward
F8=Forward F9=Swap F12=Cancel

```

Figure 44. CTC Connection Update List

Copying a channel subsystem to an SMP processor

HCD offers actions to copy a channel subsystem to an SMP processor or merge the CSS to an existing SMP processor configuration. When copying a channel subsystem to an SMP processor, the following related objects and connections are also copied:

- Partitions
 - Channel paths
 - Control units
 - Devices
1. Use action *Work with channel subsystems . . (XMP)* (or action code **s**) for an XMP processor which brings you to its Channel Subsystem List.
 2. Now you select action *Copy to processor* (or action code **y**) for the CSS that you want to copy.
 3. The Identify Target IODF panel is displayed. Specify the IODF where the target processor is defined. The default IODF is the IODF you are currently working with. Press the Enter key. The Copy to Processor panel is displayed.

```
Copy to Processor

Specify or revise the following values.

Source channel subsystem:
Processor ID . . . . . : PROC4      IBM z196 processor
Channel subsystem ID . . . : 0

Target processor:
Processor ID . . . . . : SMPP01__ +
```

4. Specify your target processor (an SMP processor) and press the Enter key.
- Note:** If the target processor does not yet exist, HCD invokes the Add Processor dialog. If the target processor already exists, you must confirm that you want to merge the configuration data. Anyway, make sure that the target processor does not support multiple logical channel subsystems.
5. Before returning to the Channel Subsystem List, a message will inform you about the success of the action.

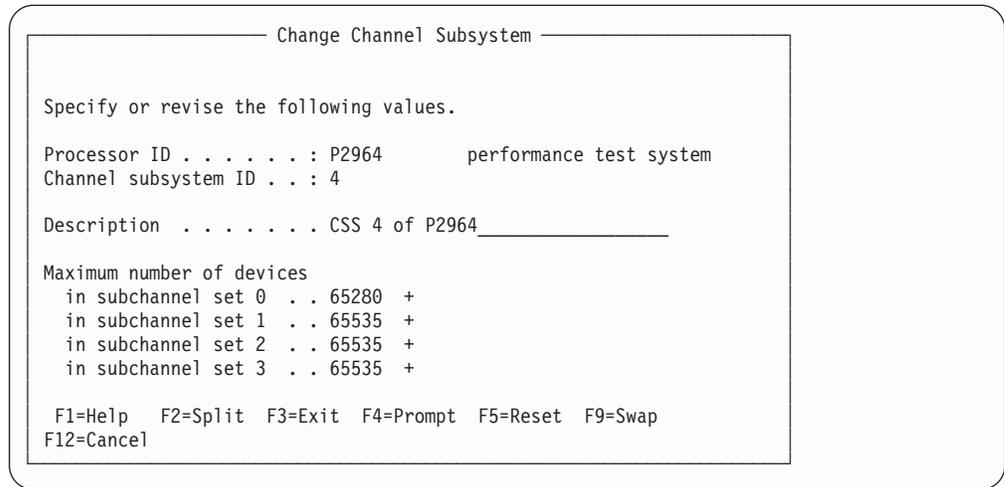
Changing channel subsystems

You can change the following characteristics of a channel subsystem:

- Description
- Maximum number of devices per subchannel set

To perform this task, proceed as follows:

1. On the Processor List, select an XMP processor and the action *Work with channel subsystems . . (XMP)* from the context menu (or action code **s**).
2. On the Channel Subsystem List, for the CSS you want to change, select action *Change* from the context menu (or action code **c**).
3. The Change Channel Subsystem panel is displayed, where you can specify your changes. Then press the Enter key. This returns you to the Channel Subsystem List.



Changing the channel subsystem ID

To change a CSS ID is only possible via the deviation of repeating (copying) the channel subsystem with a new ID and then delete the source CSS. If you cannot repeat the channel subsystem in the same processor, because all available IDs are occupied, and you want to exchange the IDs of two existing CSSs, then you need to copy both CSSs into a different target processor, delete them in the source processor and copy them back to the source processor with the exchanged IDs.

It may be necessary to restore certain definitions afterwards, for example, coupling facility connections get lost during the copy process.

Deleting channel subsystems

You can delete the definition of a channel subsystem using the *Delete* action from the context menu (or action code **d**) on the Channel Subsystem List. If you delete a channel subsystem, all channel paths, partitions, and connections to control units and devices for that CSS are also deleted; the control units and devices are not deleted.

Working with partitions

The following section describes how to work with partitions. Note that you can define partitions for a processor regardless of whether it is defined with configuration mode BASIC or LPAR.

Defining partitions

Define partitions as follows:

1. On the *Primary Task Selection* panel, select *Define, modify, or view configuration data* and on the resulting panel the object *Processors*. HCD displays the Processor List of processors currently defined in the IODF.
2. On the Processor List:
 - **for SMP processors**, select the processor and the *Work with partitions* action from the context menu (or action code **p**).
 - **for XMP processors**, select the processor and the *Work with channel subsystems . . (XMP)* action from the context menu (or action code **s**) to display the Channel Subsystem List. From this list, select the appropriate channel subsystem and the *Work with partitions* action from the context menu (or action code **p**).

HCD displays the Partition List showing the currently defined partitions for the designated processor. For most processor types HCD will populate all possible partitions as reserved partitions, when adding a processor.

```

----- Partition List -----
Goto Backup Query Help
-----
Row 1 of 15
Command ==> _____ Scroll ==> CSR

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

Processor ID . . . . : P2964V3 z13 V3 support
Configuration mode . : LPAR
Channel Subsystem ID : 0

/ Partition Name  Number Usage + UID  Description
- LP0A           A    CF/OS  Y    Linux Test
- LP0B           B    OS      Y    Linux Test BTM
- LP0C           C    OS      Y    Linux Test
- LP0D           D    OS      N    Mini-VM 5 Guests
- LP0E           E    OS      N    Mini-VM 10 Guests
- LP0F           F    OS      N    Bib-VM 60 Guests + BTM guests
- LP01           1    OS      N    Linux Test
***** Bottom of data *****

```

- Use F11=Add to add the partitions. The data-entry fields are shown in the following figure, with sample data:

```

----- Add Partition -----

Specify the following values.

Partition name . . . LP01_____
Partition number . . 1 (same as MIF image ID)
Partition usage . . OS +
UID uniqueness . . . N (Y/N)

Description . . . . Test partition 01_____

```

The partition usage field marks a partition to be used for coupling facility support or for operating system usage. The type of partition usage can be either: CF, OS, or CF/OS.

Specify CF/OS if the partition usage will be determined at partition activation. You can then include this partition into the access list of all channel path types. At partition activation those definitions are ignored that are not valid for the actual usage.

The partition UID uniqueness field can be set to 'Y' (the default is 'N') if PCIe UID uniqueness is required for PCIe functions accessed by this partition. PCIe UID uniqueness checking is done during build production IODF.

- Press the Enter key. HCD displays the updated Partition List, if you have not yet defined any channel paths.

If you have already defined channel paths, HCD displays the Update CHPID Access and Candidate Lists panel, where you can include the partition in the access or candidate list of a channel path. For an explanation of access and candidate list, refer to “Defining channel paths” on page 109.

After pressing the Enter key HCD displays the Update Device Candidate Lists panel, if the new partition is given access to a channel path that attaches devices with an explicit device candidate list. Use this panel to add the new partition to the device candidate list of the listed devices.

Note: Depending on the defined processor type there might exist one or more partitions which can not be used/changed because they are reserved for internal use.

Defining reserved partitions

For XMP processors, HCD provides the capability to add or remove logical partitions via dynamic I/O configuration. In an IODF used to create your initial IOCDS for power-on reset (POR), you can define reserved partitions, which you plan to add dynamically at a later point in time. In the *Add Partition* dialog (see step 3 on page 98 from the previous list), you specify an '*' as the placeholder partition name for reserved partitions. Reserved partitions will appear with this '*' at the end of the Partition List. Furthermore, you specify a partition number, a usage type and optionally a description.

Reserved partitions do not appear in the access or candidate lists of channel paths or devices.

To activate a partition dynamically, you need to change the '*' name to a valid partition name and to define the appropriate partition configuration before building a new production IODF.

Note: You cannot change the partition number dynamically.

Changing partitions

You can change the following data of a partition using the *Change* action from the context menu (or action code **c**) on the Partition List. On the resulting Change Partition panel, you can change:

- Name
- Number
- Usage
- UID uniqueness
- Description

If there are already channel paths attached to the partition, HCD displays the channel path access and candidate lists after pressing the Enter key on the Change Partition panel. On these lists, you can update the channel path access of the partition.

Notes:

- You can also change these partition definitions (except the name) by simply typing over the appropriate columns on the Partition List.
- To dynamically change the "UID uniqueness" attribute of a partition, the partition cannot be active.

Changing partition names dynamically

To dynamically change a partition name, you have to perform two steps:

1. Disconnect all channel paths and devices from the partition and change the partition name to * (see "Defining reserved partitions"). Activate this intermediate IODF. (This is the only required step if you want to keep this partition as a reserved partition).
2. Now you can change the * partition name to a new valid name, reconnect the wanted channel paths and devices and activate the new configuration.

Repeating (copying) partitions

You can copy partitions within the same or to another IODF. When copying a partition, the following related objects and connections are also copied:

- Channel paths having the partition in their access list
- Control units reached by the partition
- Devices reached by the partition

Note: CHID values are not copied.

Before copying the partition, perform the same checks as when repeating a processor (see “Repeating (copying) processors” on page 89). Omit step 1 that applies to a processor repetition only.

Then copy a partition as follows:

1. Copy the partition using the *Repeat (copy) partitions* action from the context menu (or action code **r**) on the Partition List. The Identify Target IODF panel is displayed.
2. Specify the IODF to which the selected partition is to be copied. The default IODF is the IODF you are currently working with.
3. On the following Repeat Partition panel, specify the required values and press the Enter key.

Definitions for the source partition are merged with the definitions of the target partition. If the target partition has another partition number than the source partition, HCD keeps the target partition number.

If control units or devices already exist in the target IODF (same number and type), HCD tries to map them. “Migrating additional IOCP input data sets” on page 283 explains in detail when a device or control unit is mapped. If they are mapped, the attributes of the target control unit or device are kept.

See “Explicit device candidate lists” on page 101 how to handle device candidate lists.

Note: HCD provides special processing when copying/repeating partitions with CTC connections. For more information refer to “Copying/repeating channel subsystems with CTC connections” on page 95.

Transferring partition configurations

Use this 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 (copy)* 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.

Before transferring the data, you must define the target channel path with its partition access and candidate list, dynamic switch ID, entry switch ID and entry port.

The new channel path may have a different type than the source channel path.

1. On the Partition List, select the *Transfer (move) partition configs* action from the context menu (or action code **x**).
2. On the *Identify Target Partition* panel, specify the target processor and partition.

3. The *Transfer Partition Configuration* panel is shown. To transfer all control units and devices reached by the source partition, specify a new CHPID value for every source CHPID. The new CHPID of the target partition must exist. The data-entry fields with sample data are shown in the following figure.

Transfer Partition Configuration Row 1 of 8

Specify the new CHPID values.

From:	To:
Processor ID . . . : PROC1	Processor ID . . . : PROC2
Partition name . . : PROD1	Partition name . . : PROD2

CHPID	Type	Mode	New CHPID +
01	FCP	SPAN	—
02	FCP	SPAN	—
03	FCP	SPAN	—
04	FCP	SPAN	—
20	CNC	SHR	—
21	CNC	SHR	—
25	CNC	DED	—
26	CNC	DED	—

4. After you press the Enter key the Partition List is displayed again. The attribute values of the transferred control units and devices remain the same for unit address/range, destination link address, time-out, and STADET. The logical address, protocol, and I/O concurrency level of a control unit remain the same if they are compatible with the target processor and channel path. If they are not compatible, default values are used.

Explicit device candidate lists

If the devices that are affected by the *Transfer (move) partition configs* action from the context menu (or action code **x**) have an explicit device candidate list, the result of the transfer action depends on whether or not the device was already connected to the target processor. See Table 6 for the different combinations. These combinations also apply to the *Repeat Partition* and *Repeat Processor* actions.

Table 6. Result of the Transfer (Move) Partition Configs Action

Transfer source partition	Transfer target partition			
	Device already connected to target partition			Device not connected to target partition
	no cand	cand +	cand -	
no cand	=	=	cand +	no cand
cand + (partition included)	=	=	cand +	cand +
cand - (partition not included)	cand - (*)	=	=	cand - (*)

Note:

Result of the Transfer Action, Relation of Device to Partition:

- = no action, target partition remains unchanged
- no cand**
no explicit device candidate list exists for partition
- cand +** partition included in explicit device candidate list
- cand –** partition not included in explicit device candidate list

Note: (*) The source partition is not included in the explicit device candidate list. During the transfer, HCD checks whether all partitions of the source processor in the candidate list have the same name on the target processor. If partitions with the same name on the target processor are identified they are added to the device candidate list for the target processor. If no partition with the same name is found for the target processor, no explicit device candidate list is built. Therefore, a partition transfer can result in a loss of candidate lists, if all partition names between source and target processor are different. It is recommended to run a device compare report after the partition has been transferred.

CF channel paths

Connections of CF sender and CF receiver channel paths will not be transferred. You have to connect them again after having transferred the partition.

Deleting partitions

You can delete the definition of a partition using the *Delete* action from the context menu (or action code **d**) on the Partition List.

To enhance deleting of partitions, which have a CHPID assigned to only this partition in either the access or candidate list, the *Confirm Delete Partition* indicates all such CHPIDs that are exclusively assigned to this partition by flagging them with an '*' (CHPIDs B8 and BA in Figure 45 on page 103). Thus, users can remove the flagged CHPIDs in one step and then delete the partition more efficiently.

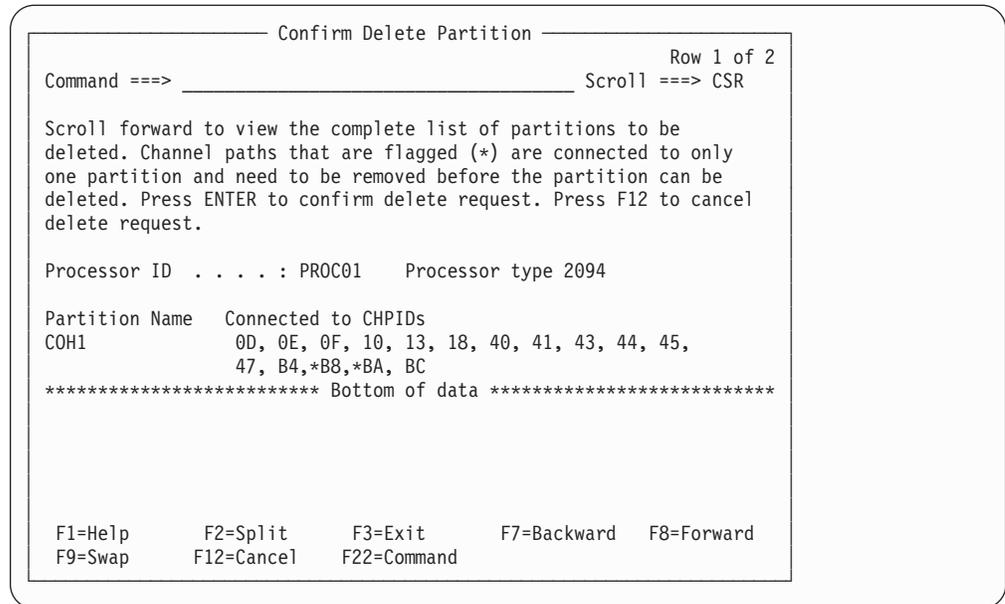


Figure 45. Confirm Delete Partition

Working with PCIe functions

Starting with processor type 2827, Peripheral Component Interconnect Express (PCIe) adapters attached to a system can provide the operating system with a variety of so-called PCIe functions to be exploited by entitled logical partitions (LPARs).

Currently HCD supports:

- **Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE).** PCIe functions of type RoCE may be assigned to external physical networks by specifying corresponding PNET IDs.
- **Internal Shared Memory PCIe Adapter (ISM).** A virtual PCIe adapter for which a virtual channel identifier (VCHID) has to be defined.
- **Regional Crypto Enablement (RCE).** This PCIe function type is used for IBM approved vendor crypto adapters.
- **zEDC-Express.** For PCIe functions of type zEDC-Express, a virtual function number between 1 and 15 must be specified.

Note: The support of virtual functions, the allowed range of virtual functions and support of PNETIDs depends on the processor type and support level. HCD offers prompts for virtual functions and ensures that the validation rules are fulfilled.

HCD provides dialogs to define, change, delete, and view PCIe functions, and to control, which LPARs have access to which PCIe functions.

Defining PCIe functions

1. On the *Primary Task Selection* panel, select *Define, modify, or view configuration data* and on the resulting panel, select *Processors*. HCD displays the Processor List of defined processors.
2. On the *Processor List* panel, select an eligible processor and action *Work with PCIe functions* from the context menu (or action code **f**).

HCD displays the *PCIe Function List* showing all PCIe functions defined for the selected processor.

```

Goto Filter Backup Query Help
-----
                    PCIe Function List      Row 1 of 15 More:  >
Command ==> _____ Scroll ==> PAGE

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

Processor ID . . . . : P2964V3      z13 V3 support

/ FID  CHID+ VF+ Type+      UID  Description
- 0010 110  1  ROCE      F010 VF and 2 pnetids, a=lp-1
- 0011 111  1  ROCE      F011 VF and 2 pnetids, a=lp01
- 0012 111  2  ROCE      F012 VF and 2 pnetids, a=lp01
- 0013 111  3  ROCE      F013 VF and 2 pnetids, a=lp01
- 001F 111 31  ROCE      F014 max VF and 2 pnetid,a=lp01
- 0020 4FE 31  ROCE      F020 max VF and 2 pnetid,a=lp01
- 0200 200  2  ZEDC-EXPRESS  ___ Zedc 200, a=lp16
- 0222 200  3  ZEDC-EXPRESS  ___ Zedc 200, a=lp16
- 0400 7C4  1  ISM          ___ ism a=lp16
- 0401 7C4  2  ISM          ___ ism a=lp16
- 0402 7C4  3  ISM          ___ ism a=lp16
- 0403 7C4  4  ISM          ___ ism a=lp16
- 0405 7C4 255 ISM          ___ ism, max vf a=lp16
- 0FF0 4FF  1  ZEDC-EXPRESS  ___ Zedc ff0,max pchid, a=lp01
- 1200 400  ___ RCE          ___ rcel200, no vf a=lp0a
***** Bottom of data *****

```

Figure 46. PCIe Function List

A PCIe function is defined by a unique identifier, the function ID (*FID*). Each function specifies a function type and a channel identifier *CHID*. Multiple functions may be specified to the same *CHID* value provided that each of these functions defines a unique virtual function *VF* number. When defining a PCIe function, you may specify a description which is shown in this list.

Use PF20=Right to scroll to the partition assignments for the displayed PCIe functions, one panel for each defined channel subsystem.

```

Goto Filter Backup Query Help
-----
PCie Function List      Row 1 of 15 More: < >
Command ==> _____ Scroll ==> CSR

Select one or more PCie functions, then press Enter. To add, use F11.
Processor ID . . . . : P2964V3      z13 V3 support
1=0S LP01      2=0S LP02      3=0S LP03      4=0S LP04      5=0S LP05
6=0S LP06      7=0S LP07      8=0S LP08      9=0S LP09      A=CO LP0A
B=0S LP0B      C=0S LP0C      D=0S LP0D      E=0S LP0E      F=0S LP0F

----- Partitions 0x -----
/ FID  CHID  VF   1 2 3 4 5 6 7 8 9 A B C D E F
_ 0010  110   1   a _ _ _ _ _ _ _ _ _ _
_ 0011  111   1   a _ _ _ _ _ _ _ _ _ _
_ 0012  111   2   a _ _ _ _ _ _ _ _ _ _
_ 0013  111   3   a _ _ _ _ _ _ _ _ _ _
_ 001F  111  31   a _ _ _ _ _ _ _ _ _ _
_ 0020  4FE  31   a _ _ _ _ _ _ _ _ _ _
_ 0200  200   2   _ c c _ _ _ _ _ _ _ _
_ 0222  200   3   _ _ _ _ _ _ _ _ _ _
_ 0400  7C4   1   _ _ _ _ _ _ _ _ _ _
_ 0401  7C4   2   _ _ _ _ _ _ _ _ _ _
_ 0402  7C4   3   _ _ _ _ _ _ _ _ _ _
_ 0403  7C4   4   _ _ _ _ _ _ _ _ _ _
_ 0405  7C4  255 _ _ _ _ _ _ _ _ _ _
_ 0FF0  4FF   1   a c c c c _ _ _ _ _
_ 1200  400   _   _ _ _ c c _ _ _ _ a c c _ _
***** Bottom of data *****

```

Figure 47. PCIe Function List

3. Use F11=Add to define a new PCIe function. The data-entry fields are shown in the following figure, with sample data:

```

----- Add PCIe Function -----

Specify or revise the following values.

Processor ID . . . . : P2964V2      z13 updated test system

Function ID . . . . . _____
Type . . . . . ROCE      +

Channel ID . . . . . 110      +
Virtual Function ID . . . . . 1      +
Number of virtual functions . 1
UID . . . . . A7B8

Description . . . . . VF and 2 pnetids, a=lp01

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

```

Figure 48. Add PCIe Function

You define a new PCIe function specified by its required function ID, function type and CHID value, the virtual function number (if possible for the specified function type), and a description. After pressing the Enter key, the LPARs can be selected that should be entitled to access the function. If you specified an

Add-like action, the data entry fields, except the function ID, and the LPAR connections are preset with the values of the selected source function.

Each PCIe function is identified by a four-digit hexadecimal function ID that is unique within a processor configuration. You must specify a function type and a CHID value which describes the related PCIe adapter card in the assigned slot of the I/O drawer. Multiple PCIe functions may be defined for the same CHID by assigning a unique virtual function ID to each of these functions.

HCD 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 dynamic activation.

You can define multiple PCIe function with identical attributes and increasing virtual function IDs at a time by specifying a value greater than 1 in the 'Number of virtual functions' field.

A UID value can be defined for a PCIe function which will be checked for uniqueness if one or more partitions in the access/candidate list have the UID flag set to yes.

Note: If partitions require uniqueness, but no PCIe UID is specified, the a unique UID is generated by HCD when a production IODF is built.

Depending on the function type, you can assign physical network IDs to a PCIe function. Each physical port of the PCIe adapter can be assigned to a (possibly different) physical network. If you press the Enter key, for a function supporting PNETIDs, HCD displays the *Add/Modify Physical Network IDs* window where you can enter a physical network ID (*PNET ID*) for each physical port of the adapter. The sequence of the PNET IDs corresponds to the sequence of the port numbers on the adapter card. All functions of a given CHID must have the same set of PNETIDs. 'Build production' will check if PCIe functions with PNETIDs specified have a corresponding channel path (same PNETID) accessing at least one identical LPAR is defined.

Add/Modify Physical Network IDs

If the Channel ID (CHID) is associated to one or more physical networks,
specify each physical network ID corresponding to each applicable physical port.

Physical network ID 1 . . PNET01 _____
Physical network ID 2 . . PNET02 _____
Physical network ID 3 . . _____
Physical network ID 4 . . _____

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

Figure 49. Add/Modify Physical Network IDs

If you press the Enter key either on the *Add/Modify Physical Network IDs* window for PCIe functions supporting PNETIDs, or directly on the *Add PCIe* function window for PCIe functions without PNETID support, HCD displays the *Define Access List* window, where you can specify one partition to be connected to the defined PCIe function.

```

Define Access List
Row 1 of 67
Command ==> _____ Scroll ==> PAGE

Select one or more partitions for inclusion in the access list.

Function ID . . . . : 222

/ CSS ID Partition Name  Number Usage Description
- 0 LP0A A OS Linux Test
7 0 LP0B B OS Linux Test BTM
.....
- 1 LP1A A OS Linux Test
- 1 LP1B B OS Linux Test BTM
.....
- 2 LP2A A OS Linux Dev
- 2 LP2B B OS Linux Dev
.....
- 3 LP3A A OS Midi test systems
- 3 LP3C C OS z/OS for HCD changes
.....
- 4 LP4D D CF/OS
- 4 LP4E E CF/OS
.....
- 5 LP51 1 CF/OS
- 5 LP52 2 CF/OS

```

Figure 50. Define Access List

Pressing Enter again leads you to the *Define Candidate List* window. Here you can modify the candidate list of partitions assigned to the PCIe function. You can define partitions from any channel subsystem.

Changing PCIe functions

You can change the following characteristics of a defined PCIe function:

- type
- CHID
- virtual function number, if supported
- UID
- description
- physical network IDs, if supported
- assigned partition in the access list
- assigned partitions in the candidate list

You can change a PCIe function using the *Change* action from the context menu (or action code **C**) on the *PCIe Function List*.

You can also change the PCIe function definitions and the related access and candidate list by simply typing over the appropriate values in the *PCIe Function List*. For certain changes of the function type, HCD enforces the specification of a virtual function number.

You have to press the Enter key to process the changes. HCD then validates the data and displays the panel again. The changes are processed sequentially row after row. Changes that require a processing of multiple row changes at once will not be possible.

Deleting PCIe functions

You can delete a PCIe function using the *Delete* action from the context menu (or action code **d**) on the *PCIe Function List*. Before deleting the PCIe function, HCD prompts you for a confirmation.

Working with channel paths

Channel paths can be dedicated, reconfigurable, shared, or spanned. The following list explains when to use which channel path operation mode.

- DED** Dedicated; if you want only one logical partition to access a channel path, specify that channel path as dedicated. You cannot reconfigure a dedicated channel path. This is the default mode.
- REC** Reconfigurable; if you want only one logical partition at a time to access a channel path and you want to be able to reconfigure the channel path from one partition to another, specify that channel path as reconfigurable.
- SHR** Shared; if you want more than one logical partition to access a channel path simultaneously, specify that channel path as shared.
- SPAN** Spanned; if in XMP processors for certain channel types, you want to have a shared channel accessed by partitions from multiple logical channel subsystems, specify that channel path as spanned.

On the Add Channel Path panel, enter a channel path type and use F4=Prompt for the operation mode to find out the allowed operation modes for the specified type.

Channel paths can also be categorized as static or managed. For more information, see “Defining managed channel paths” on page 113.

Using Multiple Image Facility

If a processor complex has Multiple Image Facility (MIF) capability, and is running in LPAR mode, multiple logical partitions can access the same shared channel paths, thereby reducing the number of required physical connections. In contrast, if a processor complex does not have MIF capability, all logical partitions must use separate channel paths to share I/O devices. For more information about LPAR mode and MIF, see the *PR/SM Planning Guide*.

More about spanned channel paths in multiple LCSSs

Depending on the processor type, in the HCD dialog you may define certain channel paths with operation mode SPAN. A spanned CHPID will have partitions belonging to more than one channel subsystem in its access and candidate list.

A spanned channel path will be created with the same CHPID number in all channel subsystems that are using it. For example, 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. Then CHPID 1A is the same CHPID in CSSs 0, 2, and 3. In CSS 1, you can use CHPID 1A for a different channel path.

Generally speaking, a channel subsystem that is not using a spanned channel can use the CHPID of that spanned channel for a separate channel path definition.

If you define a channel as SPAN, but connect it to partitions from a single channel subsystem only, then HCD displays its operation mode as SHR. The other way

round, if a shared channel path is eligible for being spanned, and you enlarge its access or candidate list with partitions from multiple logical channel subsystems, then HCD displays this channel's operation mode as SPAN.

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.

Defining channel paths

At first, you define a channel path together with its access to logical partitions. Then you may define special channel path characteristics. These possibilities are described in "Defining special channel path characteristics" on page 112.

1. On the HCD entry panel, select the task *Define, modify, or view configuration data* and from the resulting panel, select *Processors*. HCD displays the Processor List of defined processors.
2. On the Processor List:
 - for **SMP processors**, select the processor and the *Work with attached channel paths (SMP)* action from the context menu (or action code **S**).
 - for **XMP processors**, select the processor and the *Work with channel subsystems . . (XMP)* action from the context menu (or action code **S**) to display the Channel Subsystem List. From this list, select the appropriate channel subsystem and the *Work with attached channel paths* action from the context menu (or action code **S**).

HCD displays the Channel Path List showing all channel paths defined for the selected processor/channel subsystem.

```

Goto Filter Backup Query Help
-----
Channel Path List          Row 1 of 6 More:  >
Command ==> _____ Scroll ==> PAGE

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

Processor ID . . . . : P2964          test system
Configuration mode . : LPAR
Channel Subsystem ID : 4              CSS 4 of P2964

      CHID+          Dyn Entry +
/ CHPID AID/P Type+ Mode+ Sw+ Sw Port Con Mng Description
- C0    4C0   OSD   SPAN  --- --- ---      No QD80 10Gb DWx
- C1    4C1   OSD   SPAN  --- --- ---      No QD80 10Gb DWx
- C2    4C2   OSD   SPAN  --- --- ---      No QD80 10Gb DWx
- C3    4C3   OSD   SPAN  --- --- ---      No QD80 10Gb DWx
- F5    7E5   IQD   SPAN  --- --- ---      No IQD 40 kB and external Bridge
- FA    7E0   IQD   SPAN  --- --- ---      No IQD 16 kB Maximum frame size
  
```

If the Type contains three asterisks (***), the IODF channel path type is unknown to the currently used HCD.

3. Use F11=Add to add channel paths. The data-entry fields are shown in the following panel, with sample data:

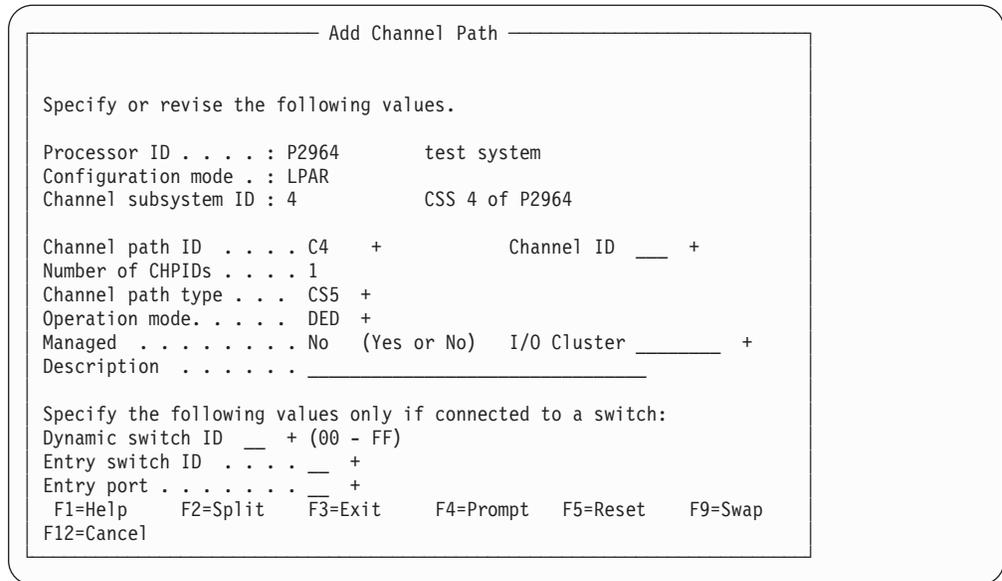
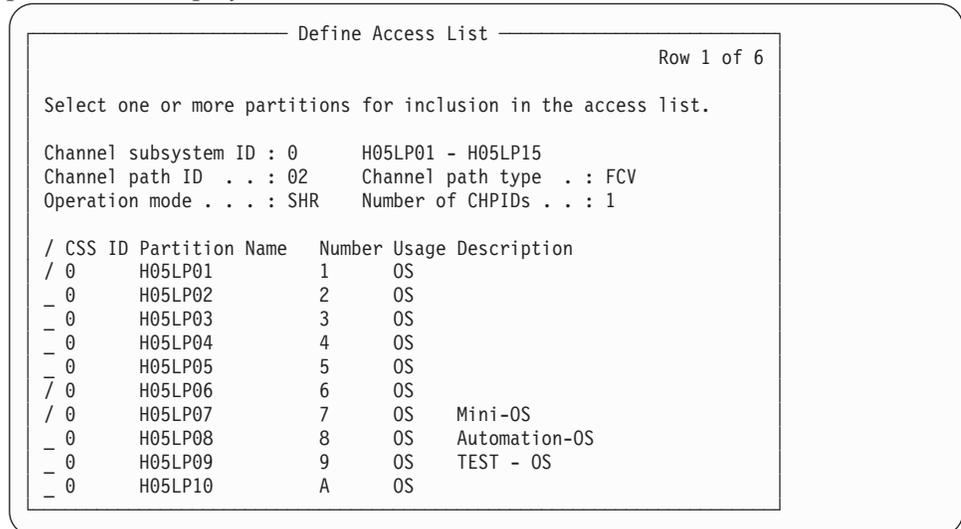


Figure 51. Add Channel Path

For physical channels on an XMP processor, you have to specify the channel identifier (Channel ID or CHID) belonging to the channel path identifier (CHPID). For internal channel paths a range of virtual CHIDs (promptable in HCD) is reserved. This range is depending on the processor type.

The CHPID Mapping Tool (CMT) can be used to make the mapping between CHPIDs easier (see “How to interact with the CHPID Mapping Tool” on page 224).

4. For each static channel path you can specify which logical partitions can access that channel path. After you press the Enter key on the Add Channel Path panel, HCD displays the Define Access List.



If you are working on *spanned* channel paths of an XMP processor, the Define Access List also shows the partitions defined for other channel subsystems:

```

Define Access List
Row 1 of 1
Command ==> _____ Scroll ==> CSR

Select one or more partitions for inclusion in the access list.

Channel subsystem ID : 0      H05LP01 - H05LP15
Channel path ID . . . : 0E   Channel path type . : OSC
Operation mode . . . : SPAN  Number of CHPIDs . . : 1

/ CSS ID Partition Name  Number Usage Description
- 0      H05LP01          1      OS
- 0      H05LP02          2      OS
...
- 1      H05LP29          E      OS      Many Guest VM
- 1      H05LP30          F      OS      TEST - VM
- 2      H05LP31          1      OS
- 2      H05LP32          2      OS
...
- 2      TRX2CFA          A      CF/OS CF for TRXPLEX
- 3      H05LP46          1      OS
- 3      H05LP47          2      OS

***** Bottom of data *****

```

If you want a logical partition to access a dedicated, reconfigurable, or shared channel path when you initially activate the logical partition, place that logical partition in the channel path's access list. For shared channel paths and spanned channel paths, you can place more than one partition in the access list.

5. If you do not include all partitions in the access list, you are prompted for the candidate list (for reconfigurable and shared channel paths) after pressing the Enter key.

From the IOCP point of view, the channel path candidate list includes the channel path access list. From the HCD point of view, the channel path candidate list does not include the channel path access list. The partitions already in the access list do not appear in the candidate list.

```

Define Candidate List
Row 1 of 12
CBDPCH1D
Command ==> _____ Scroll ==> CSR

Select one or more partitions for inclusion in the candidate list.

Channel subsystem ID : 0      H05LP01 - H05LP15
Channel path ID . . . : 02   Channel path type . : FCV
Operation mode . . . : SHR   Number of CHPIDs . . : 1

/ CSS ID Partition Name  Number Usage Description
- 0      H05LP02          2      OS
/ 0      H05LP03          3      OS
/ 0      H05LP04          4      OS
/ 0      H05LP05          5      OS
- 0      H05LP08          8      OS      Automation-OS
- 0      H05LP09          9      OS      TEST - OS
- 0      H05LP10          A      OS
- 0      H05LP11          B      OS
- 0      H05LP12          C      OS
- 0      H05LP13          D      OS      Mini-OS 2
F1=Help      F2=Split      F3=Exit      F5=Reset      F6=Previous
F7=Backward  F8=Forward   F9=Swap      F12=Cancel   F22=Command

```

If you want to be able to configure a reconfigurable or shared channel path online to a logical partition, place that logical partition in the channel path's candidate list.

- After pressing the Enter key, you return to the Channel Path List. Scroll to the right to get an overview of the access and candidate list of a channel path. The following matrix is displayed:

```

Channel Path List          Row 1 of 78 More: <
Command ==> _____ Scroll ==> CSR

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

Processor ID : PROC03      CSS ID : 0      H05LP01 - H05LP15
1=OS H05LP01  2=OS H05LP02  3=OS H05LP03  4=OS H05LP04  5=OS H05LP05
6=OS H05LP06  7=OS H05LP07  8=OS H05LP08  9=OS H05LP09  A=OS H05LP10
B=OS H05LP11  C=OS H05LP12  D=OS H05LP13  E=OS H05LP14  F=OS H05LP15
      CHID          I/O Cluster
/ CHPID AID/P Type+ Mode+ Mng Name + ----- Partitions 0x -----
_ 0D    100  OSN   SPAN No  _____ 1 2 3 4 5 6 7 8 9 A B C D E F
_ 10    101  CNC   SHR No  _____ c a a a a a a a a a a a a a
_ 11    108  CNC   SHR No  _____ c -----
_ 12    109  CNC   SHR No  _____ c c c c c c c c c c c c c c a
_ 13    110  CNC   SHR No  _____ c -----
_ 14    111  CNC   SHR No  _____ c -----
_ 15    170  CNC   SHR No  _____ ----- a -----
_ 16    172  CTC   SHR No  _____ c -----
...
_ 23    1F0  CNC   SHR Yes UTCPLX38 * * * * ----- * -----
_ 24    1F1  CNC   SHR Yes UTCPLX38 * * * * ----- * -----
_ 25    16/1 CIB   SHR No  _____ a c c c c c c -----

```

Figure 52. Channel Path/Partition Matrix

The legend, which precedes the partition matrix shows how the partition names are associated with the columns of the partition matrix. The headings *Partitions 0x*, *Partitions 1x*, *Partitions 2x* and so on, if scrolling to the right, indicate that the partitions for the related channel subsystems (0, 1, 2, ...) are shown. The column numbers correspond to the partition numbers in the pertaining channel subsystem. Also, the partition usage type OS, CF or CO (for CF/OS) is indicated in the legend.

In the previous example, column 1 under Partitions 0x shows the definitions for partition H05LP01 of usage type OS with partition number 1 in CSS 0.

The following entries may appear in the partition matrix:

- a** indicates that the partition is in the channel path's access list.
- c** indicates that the partition is in the candidate list.
- *** is shown for a managed channel path in all logical partitions that potentially can access that channel path.
- #** indicates that the channel path (which is defined to the channel subsystem named in the Channel Subsystem ID field) cannot be attached to the partitions of another channel subsystem. Either a channel path with the same identifier is already defined for the other channel subsystem, or the channel path cannot be spanned or it can be spanned, but the channel path mode is not SPAN or SHR. For information on how to change a CHPID's operation mode to SPAN, if applicable, refer to "Changing the operation mode of a channel path" on page 122.

Defining special channel path characteristics

This section handles the following topics:

- "Defining managed channel paths" on page 113
- "Defining multiple channel paths in one step" on page 113

- “Connecting a channel path to a switch”
- “Defining IQD channel parameters” on page 114
- “Defining more than 160 TCP/IP stacks” on page 115
- “Defining an OSD channel path to physical networks” on page 115
-
- “Defining or editing channels using Host Communication Adapters:” on page 115
- “Defining spanned channel paths” on page 116
- “Over-defining a CHPID” on page 116

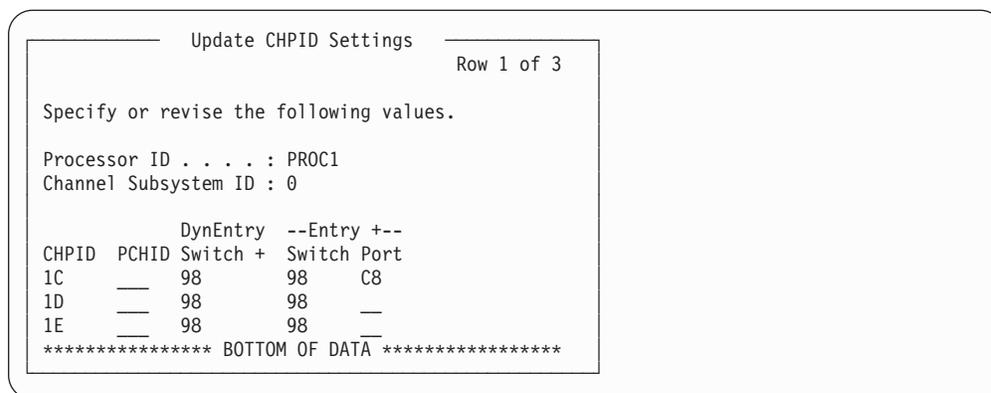
Defining managed channel paths: You can define a channel path as being managed by Dynamic Channel Path Management (DCM). DCM will use such a channel path to dynamically assign the logical paths to control units in order to optimize I/O activity. A managed channel path must connect to a dynamic switch and may be used for control units that connect to the same switch. If a channel path is defined as managed in an LPAR mode processor, it must be defined as shared. It cannot be connected to logical partitions but must specify an I/O cluster name. An I/O cluster is a sysplex that owns the managed channel path. All systems of the sysplex on the given processor are allowed to share the managed channel path. A managed channel path cannot be connected to a control unit by HCD.

Defining multiple channel paths in one step: You can define, in one step, a group of channel paths of the same type and mode and with consecutive identifiers. It is recommended to define only a group of channel paths that have the same partitions in their access and candidate lists. Otherwise, you have to change the channel paths that have different partitions in their access and candidate list in a further step.

1. Define the group by specifying the first channel path identifier (CHPID) and the number of channel paths in the group. Define a channel path type, mode, and description. HCD applies the definition to all channel paths in the group.
2. Type over the fields that are different, for example description, in the Channel Path List.

Connecting a channel path to a switch: If you have already defined a switch, you can connect the channel path to the switch on the Add Channel Path panel. Specify the dynamic switch ID, the entry switch ID, and the entry port to connect the channel path to a switch.

The values are only valid for the first channel path if you have defined a group of channel paths in one step. To define values for the other channel paths of the group, HCD displays an additional panel. This panel allows you to define the entry ports for all subsequent channel paths of the group. For information on dynamic switch ID and entry switch ID, refer to “Possibilities of switch connections” on page 163.



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

Maximum frame size

If you define or update an IQD channel path, HCD displays a dialog that 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) support offers two new options besides the basic HiperSockets functions:

- 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)

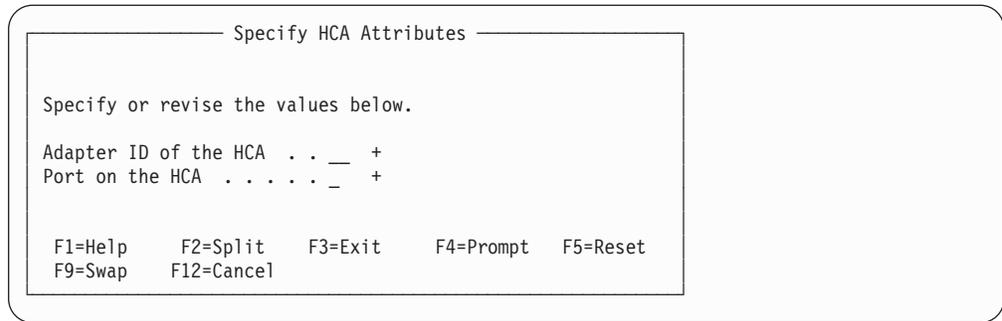
The IQD channel path supports IEDN via the Internal Queued Direct I/O Extensions (IQDX) function.

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.

Physical network ID

An IQD channel path may be defined to a physical network. If the channel path is defined with IQD function Basic HiperSockets or External Bridge, a customer network can be specified. An IQD channel path with IQDX function is defined to the internal network IEDN, per default.



Defining spanned channel paths: You can define a suitable channel path as *spanned* directly when creating it. On the Add Channel Path panel from Figure 51 on page 110, specify

Operation mode SPAN

After pressing the Enter key, HCD displays the combination of the Define Access List and the Define Candidate List, offering partitions from multiple logical channel subsystems. Note that if you, nevertheless, in both lists select only partitions from the current CSS, then the operation mode of the channel path is set back to SHR.

For information on how to change a CHPID's operation mode to SPAN, if applicable, refer to "Changing the operation mode of a channel path" on page 122.

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 CHID 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 CHID 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 CHID value or HCA ID *, the whole control unit (including any attached devices) is omitted from the configuration.

If a CHPID changes its CHID or HCA ID from * to a valid value during a dynamic activation, an 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 CHID/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 CHID=*, VCHID=*, 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.

Establishing coupling facility channel path connections

Before you start to establish a Coupling Facility (CF) channel path connection, you must have defined a processor that supports coupling facilities, a coupling facility partition, a coupling facility receiver (CF receiver) channel path and coupling facility sender (CF sender) channel path, or peer channel paths.

1. On the Channel Path List (Figure 52 on page 112) select a channel path and the **Connect CF channel path** action from the context menu (or type action code **f**). HCD displays the **CF Channel Path Connectivity List** showing all CF channel paths defined for a processor.

To show that a CHPID is already connected in another IODF you can set the indicator in the Occ (occupied) column to Y (yes). You cannot connect a CHPID labeled Y. However, you can change the occupied status by overwriting.

```

                                CF Channel Path Connectivity List                                Row 1 of 6
Command ==>> _____ Scroll ==>> CSR

Select one or more channel paths, then press Enter.

Source processor ID . . . . . : R35          Proc type 2817 (z196)
Source channel subsystem ID . : 0
Source partition name . . . . . : *

-----Source-----      -----Destination-----      -CU-  -#-
/ CHP CHID CF Type Mode Occ Proc.CSSID CHP CHID CF Type Mode Type Dev
- 02 00/1 Y CIB SHR N      H05.3 0E 518 Y CFP SHR CFP 7
- 03 01/1 Y CIB SHR N
- 0E 1E1 N CFP SPAN N      H05.3 0F 528 Y CFP SHR CFP 7
- 0F 1E9 N CFP SPAN N      ECL2.0 81 201 Y CFP SPAN CFP 7
- F7 270 Y CFP SPAN N      H05.3 06 519 Y CFP SHR CFP 7
- F8 271 Y CFP SPAN N      ECL2.0 85 211 Y CFP SPAN CFP 7
- F9 278 Y CFP SPAN N      H05.3 07 529 Y CFP SHR CFP 7
- FA 279 Y CFP SPAN N
***** Bottom of data *****

```

Figure 53. CF Channel Path Connectivity List

Note:

- a. The *Source partition name* field indicates a name only when the Filter function is employed.
- b. Column *CHID* for both the source and destination of the connection, is either the physical channel identifier to which the channel path is assigned or, for some CHPID types, its host communication adapter ID and port number.
- c. Column *CF* indicates **Y** if at least one partition specified in the access or candidate list is of type CF or CF/OS, which is a prerequisite for establishing CF channel path connections.
- d. Column *CU Type* indicates the type of the connected control unit(s).
- e. Column *# Dev* indicates the number of CF devices connected to the CU(s) and used for the CF connection.

2. Select the source channel path for a coupling facility connection and the *Connect to CF channel path* action from the context menu (or action code **p**). HCD displays the Connect to CF Channel Path panel. The data-entry fields are shown in the following figure, with sample data:

Connect to CF Channel Path

Specify the following values.

```

Source processor ID . . . . . : XMPPROC1
Source channel subsystem ID . . : 0
Source channel path ID . . . . . : 1B
Source channel path type . . . . : CFP

Destination processor ID . . . . . XMPPROC2 +
Destination channel subsystem ID . . 0 +
Destination channel path ID . . . . 1B +

Timing-only link . . . . . . . . . . No

```

Figure 54. Connect to CF Channel Path

3. To establish the CF channel path connection, specify the destination processor ID, destination channel subsystem ID and destination channel path ID.
 If the CF control unit definition does not yet exist, HCD automatically generates a CF control unit and CF devices for a sending channel path when CF channel paths are connected. HCD uses type CFS for a (legacy) CF sender channel path and type CFP for a CF peer channel path. The sending CF channel paths, that connect one CSS of a processor to a CF partition, are assigned to the same CF control unit. For each CF sender channel path connection, HCD generates two CF devices. For each sending CF peer channel path, HCD generates seven (or 32) CF devices. This allows eight CF links between a single CSS and a specific target CF partition. HCD proposes the highest unused control unit number and highest unused consecutive device numbers in the IODF. If a CF peer channel path does not connect to a target CF partition (that is, the sending function is not used), HCD does not connect the channel path to CF control unit and CF devices.

If a CF control unit definition is already used for another CF connection from the CSS of the sending CF channel path to the same target CF partition, HCD proposes the same control unit number. This control unit number may be overwritten by an unused control unit number, provided the partition lists of the channel paths on the existing control unit do not overlap with the partition list of the sending CF channel path for the new CF connection. Thus, it is possible to establish more than eight CF links between a single CSS of a CPC and a specific target CF partition.

Note: HCD checks the partition access lists of the channel paths for an overlap. That means, you can define an overlap in the partition candidate lists. In such cases however, you must ensure by operational means that at any one time, the CHPIDs of only one of the control units with overlapping partitions are configured online.

You can also choose to use the same control unit for all partitions of a sysplex that is shared across more than one CSS of a processor for the CF links to a specific target CF partition. This is necessary, if you plan to use a mix of shared

(SHR) and spanned (SPAN) CIB connections between your syplex and the target CF partition. In such a case, it is recommended to define first the SPAN and then the SHR CHPIDs.

Note: If you start defining SHR CHPIDs from two CSSs connected to the same target CF partition, HCD proposes for each CSS its own control unit with the CHPIDs from its CSS. If you now want to add a connection with a spanned CHPID that is spanning both of the original CSSs, you cannot add the new spanned CHPID to both control units. Instead, you must break all the connections from one CSS, then connect the new spanned CHPID, with HCD forcing you to use just one control unit, and then reconnect all the broken CHPIDs, with HCD also forcing them to use the same control unit.

The *Add CF Control Unit and Devices* panel is displayed twice (for the source and for the destination side), where you must confirm or revise the values for the CF control unit and CF devices. The data-entry fields are shown in Figure 55, with sample data:

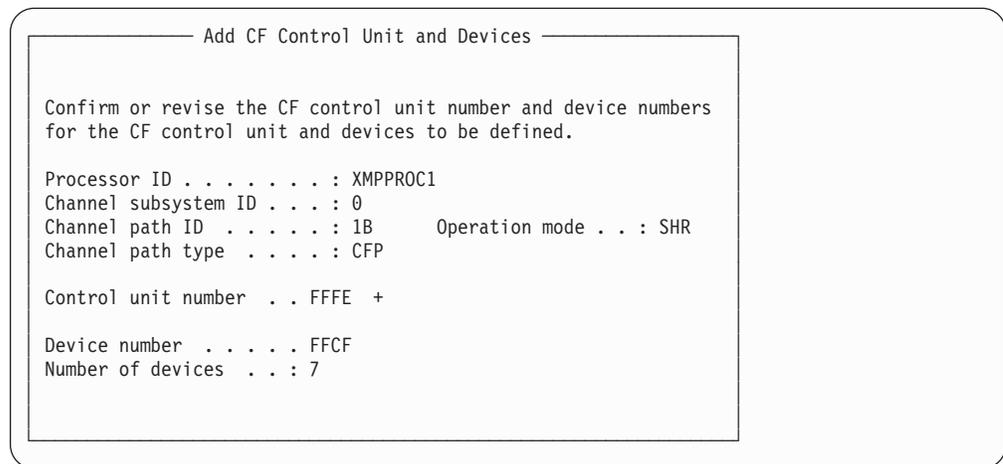


Figure 55. Add CF Control Unit and Devices

Note:

- a. The CF control unit and device definitions are displayed on the Control Unit List and on the I/O Device List, but in a disabled state where they cannot be modified or deleted.
- b. If you specified a timing-only link in the dialog from Figure 54 on page 118, then the field *Number of devices* is set to 0 and cannot be changed, as no devices are created for such links. For more information on timing-only links (STP links), read “Defining Server Time Protocol (STP) links” on page 120.

- c. If you are defining a coupling facility link (CF link) the field *Number of devices* is initially defaulted according to the chpid and processor type. If both processors connected by the CF link, support more subchannels than initially set the *Number of devices* value can be changed.

Changing the *Number of devices* results in a new device number (except you changed this value as well; in this case a message is shown) and a redisplay of the window showing the modified values. Pressing Enter again displays the same window again, but this time with the destination side values.

4. After you press the Enter key, HCD redisplay the CF Channel Path Connectivity List with the new connection defined.

Defining Server Time Protocol (STP) links

HCD supports Server Time Protocol (STP) links (timing-only links) between two zSeries (z890, z990) or later processors. Timing-only links are only needed in the case when coupling links are neither desired nor possible or not needed. If you want to define a coupling facility connection which will be used as a timing-only link, you must set the *Timing-only link* entry in the *Connect to CF Channel Path* panel (Figure 54 on page 118) to Yes. Both source and destination processors must be timing capable in this case, and the used channel paths must be from one of the channel path types CFP, CBP or CIB.

Establishing a timing-only link between two processors does not require a CF partition, but can be established between two OS partitions.

For an STP only link, HCD generates a control unit of type 'STP' on both sides of the connection. No devices are defined. 'STP' is used as control unit type in the *CF Channel Path Connectivity List* in column *CU type*, which indicates the type of the connecting control unit(s) for non-STP links (see Figure 53 on page 117).

For changing a CF connection to an STP only connection and vice versa, you must break the existing connection and establish a new one.

Disconnecting coupling facility channel path connections

Perform the following steps to break a coupling facility channel path connection:

1. On the Channel Path List select any channel path and the *Connect CF channel path* action from the context menu (or action code **f**) HCD displays the CF Channel Path Connectivity List showing all CF channel paths defined for a processor.
2. Select the source channel path for a coupling facility connection and the *Disconnect* action from the context menu (or action code **n**).

Note: The appropriate CF control unit definition is removed implicitly with the last broken connection to the coupling facility to which the control unit belongs. The appropriate CF device definitions are removed implicitly, when the coupling facility connection to which they belong is broken.

Changing channel paths

To change channel path data you have to follow the same panel flow as for defining channel path data:

- Changing channel path characteristics
- Changing channel path access and candidate list

The following steps describe the panel flow and where you can change the data.

1. On the Channel Path List, select a channel path and the *Change* action from the context menu (or action code **c**).
2. On the following Change Channel Path Definition panel you can change channel path definitions such as:
 - Channel path ID (see also "Changing processors" on page 87 for an example of the Update Channel Path Identifiers panel, and how to change the CHPID values)
 - Channel path type (see "Changing the type of a channel path" on page 121)
 - Operation mode (see "Changing the operation mode of a channel path" on page 122)

- CHID
- Description
- If connected to a switch
 - Dynamic switch ID
 - Entry switch ID
 - Entry port

3. After pressing the Enter key, the Define Access List is displayed. Select one or more partitions to be included in the access list.
4. After pressing the Enter key again, the Define Candidate List is displayed (if applicable). Select one or more partitions to be included in the candidate list.

You can also change channel path definitions (except the channel path ID) and the channel path's access and candidate list by simply typing over the appropriate values on the Channel Path List. To change the access and candidate list definitions, scroll to the right to see the channel path/partition matrix (refer to Figure 52 on page 112). Overwrite the values in the channel path/partition matrix with either a for access list and c for candidate list.

Changing the ID of a channel path

Changing the ID of a channel path may first require the disconnection of the entry switch and entry port on the channel path. If channel paths of multiple processors or channel subsystems (e.g. spanned CHPIDs) connect to the same entry switch and entry port, proceed as follows:

1. Remove the entry switch and entry port from the channel path definitions.
2. Change the channel path ID of the corresponding channel paths.
3. Once again add the entry switch and entry port to the channel paths.

Changing the type of a channel path

Changing the type of a channel path from parallel to serial (or vice versa) will result in changing the type of all other channel paths that are attached to the affected logical control units. When changing the type of a channel path:

- The new/changed channel path type must not conflict with the already existing channel path IDs, control unit and device parameters. Adjust the values of the affected control units and devices according to the rules of parallel or serial channel path type. (For information on how to change control unit processor attachment and device parameters, see: "Changing control units" on page 130.)
- When changing from serial to parallel, you have to disconnect the entry switch and entry port first, if the channel path is connected to a switch.
- When changing a channel path of type BL or BY that is connected to more than one control unit, the channel path has to be defined to a corresponding CVC (converter channel path) first and then be changed to serial.

Note: A channel path type change cannot be performed in one step while changing the channel path ID.

Changing the type of a coupling facility channel path: To change the type of CF channel paths, disconnect the channel path you want to change before performing the type change. Any coupling facility devices associated with the changed channel path are removed by HCD. The associated control unit is removed only when the last connection to the coupling facility to which the control unit belongs is broken. (For details on that task, see "Establishing coupling facility channel path connections" on page 117.)

Changing the operation mode of a channel path

Changing the operation mode of a channel path is dependent on its type. For example, BL, BY, CVC, CBY, and CF receiver channel paths cannot be shared.

Before you can change the operation mode of a channel path, the rules for partition access and candidate lists of those channel paths that are attached to the affected logical control units must conform to the rules for the new operation mode. You have to check which partitions have access to these channel paths. When changing the channel path operation mode from SHR to REC or DED, you first have to remove partitions in the appropriate access and candidate lists. The partition lists for the affected logical control units have to be changed when the mode change has been done.

Changing the operation mode of a channel path to SPAN: If you want to change the operation mode to SPAN for applicable channel path types, you must ensure that the CHPID is unused in those channel subsystems into which it should be spanned. This means that the CHPID whose operation mode you want to change, must be unique throughout the processor complex. So you need to distinguish the following scenarios:

- For an existing shared CHPID that is uniquely defined throughout all LCSSs of the processor:

Enlarge its access and candidate lists with partitions from other channel subsystems. For a shared CHPID, in the Channel Path List, scroll right once for each CSS of the current processor to see the available partitions that you can specify for access or candidates.

```

Goto Filter Backup Query Help
-----
Channel Path List          Row 1 of 1 More: < >
Command ==> _____ Scroll ==> CSR

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

Processor ID : XMPP01      CSS ID : 0      LPAR01 - LPAR02
1=OS LPAR01  2=OS LPAR02  3=          4=          5=
6=          7=          8=          9=          A=
B=          C=          D=          E=          F=
          CHID          I/O Cluster  ----- Partitions 0x -----
/ CHPID AID/P Type+ Mode+ Mng Name +      1 2 3 4 5 6 7 8 9 A B C D E F
_ 00      ___ IQD  SPAN No      _____ a c -----
***** Bottom of data *****

```

1=OS LPAR01 **2=OS LPAR02** and **----- Partitions 0x -----** in our example indicate that partitions **LPAR01** and **LPAR02** are available as partitions of usage type OS in the CSS with ID=0 (0x).

With the codes **a** and/or **c** you specify which partitions you want to access. Note that if the CHPID's operation mode in column *Mode* of the Channel Path List was SHR before your changes, it is set to SPAN automatically after specifying partitions from different channel subsystems (0x, 1x, ...).

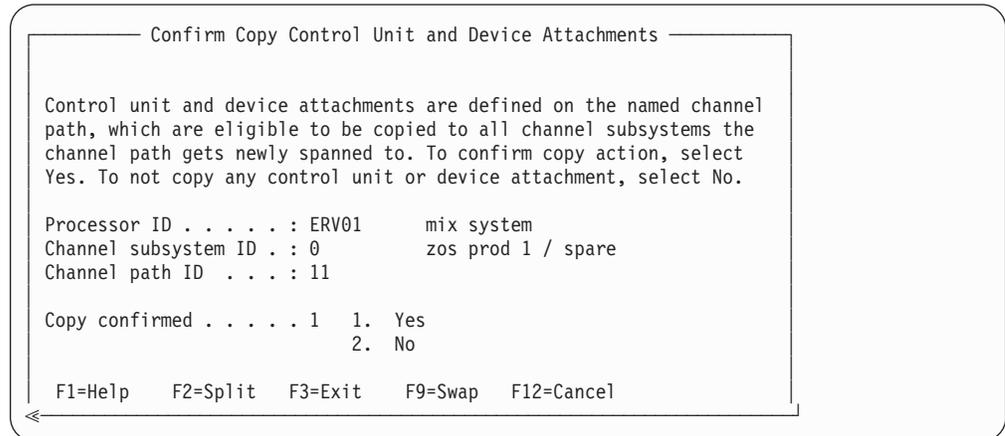
- For an existing dedicated or reconfigurable CHPID that is uniquely defined throughout all LCSSs of the processor:

You can change its operation mode to SPAN using the *Change* action from the context menu (or action code **c**) on the Channel Path List. In the subsequent Define Access List and Define Candidate List, you must select at least two partitions from different channel subsystems, because otherwise, HCD sets the operation mode to SHR.

- For any existing CHPID that is multiply defined throughout the LCSSs in the processor complex:

You must delete the CHPID from all but one LCSS, before you can change its operation mode to SPAN, using the *Change* action from the context menu (or action code **C**) on the Channel Path List and selecting appropriate partitions from the Define Access List and the Define Candidate List.

When spanning a channel path, that has control unit(s) (and devices) attached, to a new CSS, HCD invokes a dialog asking whether these control unit(s) (and devices) should also be reachable from the new CSS.



Specify Yes, if you want all existing CU and device connections of the designated channel path to be copied to all channel subsystems the channel path gets newly spanned to.

Changing the operation mode of CF channel paths: CF channel paths that connect a processor to the same coupling facility partition via the same control unit must be either all shared or all nonshared. Hence, if you want to change the operation mode for one channel path (from non-SHR to SHR), you have to change it for all. To do this proceed as follows:

1. Disconnect all receiving CF channel paths that connect a processor to the same coupling facility partition (see “Disconnecting coupling facility channel path connections” on page 120).
2. Change the operation mode of all sending CF channel paths.
3. Re-establish the connections for all channel paths that you disconnected in step 1.

The associated coupling facility control unit and coupling facility devices are removed and generated again by HCD.

Changing a coupling facility connection

To change a coupling facility connection:

1. Disconnect a connection (see “Disconnecting coupling facility channel path connections” on page 120).
2. Establish the new connection (see “Establishing coupling facility channel path connections” on page 117).

Note: A Y (yes) in column Occ (occupied) of the channel path list indicates that this CHPID is occupied. This is useful if you have another IODF where the CF connection is already defined. If you wish to connect this path, you must first change the occupied status by overwriting the Y with N.

It is not possible to include the first CF or CF/OS LP in, or remove the last CF or CF/OS LP from, the combined access and candidate list of a connected CF peer channel. Instead, the CF connection first has to be removed. Then, the CF LP can be connected to or disconnected from the CF peer channel path; afterwards, the CF peer connection can be reestablished. This is to avoid implicit generations of deletions of CF control units and devices.

Aggregating channel paths

When selecting *Aggregate channel paths* from the Channel Path List against a channel path, HCD offers the possibility of moving all control units from a source channel path to the selected target channel path of the same processor. This is useful for combining several under-utilized channel paths to a single one.

You can aggregate channel paths using the *Aggregate channel paths* action. The following steps describe the procedure:

1. On the Channel Path List select a channel path and then the *Aggregate channel paths* action from the context menu or type action code **g** next to the selected CHPID.
2. On the following Aggregate CHPID definition panel you can enter the target channel path ID for the aggregate action.

Aggregate CHPID

Specify a CHPID to be aggregated with the selected target CHPID.

Processor ID : PROC1
Channel subsystem ID : 0
Channel path ID . . : 00
Channel path type . : CNC

CHPID to aggregate . . . __ +

HCD displays a list of control units that are currently attached to the source CHPID. If possible, each control unit shows the switch port to which it is connected. Also, the target switch port and the target link address after the aggregate is shown if HCD can determine these. You can select all or a subset of control units to be aggregated to the target CHPID. The selected control units and their attached I/O devices are disconnected from the source CHPID and connected to the target CHPID. The target CHPID may now be connected to a different switch than the source CHPID. Panel *Select Control Units to be Aggregated* allows you to change the control unit port and link address for the move to the target CHPID.

```

Select Control Units to be Aggregated
Row 1 of 12
Command ==> _____ Scroll ==> PAGE

Select control units to be aggregated to target channel path. If none is
selected, no aggregation will take place.

--- Control unit -- -- Source --- ----- Target -----
/ Num Type          Switch Port   Switch Port   Linkadd
- 0000 2107            50    09    50    09    6009
- 0010 2107            50    10    50    10    6010
- 0020 2107            50    11    50    11    6011
- 0030 2107            50    12    50    12    6012

F1=Help F2=Split F3=Exit F4=Prompt F5=Reset
F7=Backward F8=Forward F9=Swap F12=Cancel F22=Command

```

After successful aggregation, a message will be displayed and the Aggregate CHPID panel will remain to allow you to aggregate additional channel paths. If aggregation fails because of validation errors, the validation errors are displayed. If prompting for channel paths for aggregation, HCD will only show the channel paths that allow aggregation without validation errors. Prompting is thus a useful planning aid.

The **Aggregate Channel Paths** action is only possible if the following prerequisites are fulfilled:

- Source and target channel paths must be different.
- All selected control units connected to the source channel path must be connectable to the target channel path.
- The source channel path must not be connected to a control unit which is already connected to the target channel path. In addition, a link address - unit address - CUADD combination used by a control unit connected to the source channel path must not also be used by a control unit connected to the target channel path.
- Either the source channel path must have the same channel path mode as the target channel path, or all devices accessible by the source channel path must be connected to only one channel path.
- Source and target channel paths must have defined a dynamic switch.
- The user must not lose connectivity by a channel path aggregate action. The source channel path access and candidate list must be the same as or a subset of the target channel path access and candidate list.
- By connecting control units of the source channel path to the target channel path, the defined maximum value for the target channel path type (e.g. maximum number of unit address ranges) must not be exceeded.

As a result of an aggregation action, HCD will:

- Change the preferred channel path of a device to the target channel path if the source channel path was the preferred channel path of the device initially.
- Leave the reachability of devices by logical partitions unchanged.
- Move the CTC control units of the source channel path port to the entry port to which the target channel is connected.

Deleting channel paths

You can delete the definition of a channel path using the *Delete* action from the context menu (or action code **d**) on the Channel Path List. If you delete a spanned channel path, it is removed from all channel subsystems which had access to it.

Working with control units

The following section describes how to work with control units.

Defining control units

You need two steps to define a control unit:

- Define the control unit characteristics
- Define how the control unit is attached to processors.

Before you define a control unit, you should have defined the processors and channel paths to which the control unit is to be attached.

Defining the control unit characteristics

1. On the *Primary Task Selection* panel, select *Define, modify, or view configuration data* and on the resulting panel the object *Control units*. HCD displays the Control Unit List showing all control units currently defined in the IODF.

```
Goto Filter Backup Query Help
-----
Control Unit List                               Row 145 of 339
Command ==> _____ Scroll ==> CSR

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

---#---
/ CU  Type +      CUADD CSS MC  Serial-# + Description
- 5E00 2105      6    5    28641  SUBCSE - S/N 28641
- 5F00 2105      7    5    28641  SUBCSE - S/N 28641
- 6000 2107      0    5    54321  SUBSQ02 - S/N 54321
- 6100 2107      1    5    54321  SUBSQ02 - S/N 54321
- 6200 2107      2    5    54321  SUBSQ02 - S/N 54321
- 6300 2107      3    5    54321  SUBSQ02 - S/N 54321
- 6400 2107      4    5    54321  SUBSQ02 - S/N 54321
- 6500 2107      5    5    54321  SUBSQ02 - S/N 54321
F1=Help   F2=Split   F3=Exit   F4=Prompt  F5=Reset   F7=Backward
F8=Forward F9=Swap    F10=Actions F11=Add    F12=Cancel F13=Instruct
F22=Command
```

Column CUADD shows the CUADD value defined for the control unit, where available. If the CUADD is inconsistently set for the control unit among processors, an * is displayed.

Column #CSS shows the number of channel subsystems to which a control unit is connected. This column contains a value only if a connection exists.

Column #MC shows the greater of the number of managed channel paths defined for the connected processors or the number of managed channel paths defined for the selected processor when coming down from the processor. This column contains a value only if managed channel paths are defined for the control unit.

Note: The CF control units generated when connecting CF channel paths are listed but are disabled for any action. It is not possible to add a new such control unit via this dialog.

- Use F11=Add to define a new control unit. The preceding data-entry fields are shown, with sample data:

Add Control Unit

Specify or revise the following values.

Control unit number 0001 +
 Control unit type 3990-3_____ +

Serial number _____
 Description DASD control unit_____

Connected to switches . . . _ _ _ _ _ _ _ _ _ _ _ _ +
 Ports _ +

If connected to a switch:

Define more than eight ports . . 2 1. Yes
 2. No

Propose CHPID/link addresses and
 unit addresses 2 1. Yes
 2. No

Defining switch connections

The *Add Control Unit* panel can also be used to specify the switches and ports the control unit is attached to.

If you specify Yes for Define more than eight ports, the *Define Control Unit Ports* dialog will be displayed to allow you to specify up to 64 control unit/switch port connections. To connect a unit to a maximum of 128 switch ports, in this dialog, you can invoke another panel to define an additional 64 switch port connections.

If you specify Yes for Propose CHPID/link addresses and unit addresses and the control unit is connected to at least one switch, HCD suggests control unit to processor attachment parameters (channel path/link addresses and the unit address range) based on the switch/ports the control unit is connected to. HCD will propose up to eight channel path/link address pairs, starting with the channel path that has the lowest number of devices attached to it.

If you add a new control unit (via Add or Add-like), HCD automatically assigns as many logical paths as possible for all processors defined.

The following prerequisites must be fulfilled for this function:

- The control unit must support ESCON or FICON attachments and not be used for channel-to-channel (CTC) connections.
- The control unit must have physical switch / port connections (switch entry ports) defined.
- Channel paths that use the connected switch as a dynamic switch must exist.

HCD then automatically selects the channel paths and link addresses according to the following rules for each processor that has been defined.

- All channel paths that use a switch that connects to the control unit as a dynamic switch are candidates for assignment.
- The channel paths are sorted ascending by the value of the reversed channel path ID. The resulting sequence is again sorted ascending by the number of connected devices.
- The connected control unit ports are ordered ascending by the numbers of already connected control units and path connections, respectively.
- For each connected switch port in the resulting sequence, the channel paths are tested in sequence. If the switch port can be used as a link address, the CHPID/link address is taken.
- A maximum number (up to 8) of possible CHPID/link address combinations is assigned.

On the following *Select Processor/Control Unit* panel you can type over the fields that are different from the suggested attachment values.

Defining processor attachment data

1. After pressing the Enter key on the Add Control Unit panel HCD displays a list that shows all the defined processors. You can then define how the control unit is to be attached to one or more processors.

A Y for Yes in the Att column indicates that the control unit is attached to the

```

Select Processor / CU      Row 3 of 5 More: >
Command ==> _____ Scroll ==> CSR

Select processors to change CU/processor parameters, then press Enter.

Control unit number . . : 0000      Control unit type . . . : 3422

-----Channel Path ID . Link Address + -----
/ Proc.CSSID 1----- 2----- 3----- 4----- 5----- 6----- 7----- 8-----
- XMPP01.0   09.0122 14.01A2 46.0222 52.02A2 33.0422 84.04A2 _____
- XMPP01.1   33.0422 84.04A2 0A.0522 14.05A2 _____
- XMPP01.2   33.0422 09.0722 14.07A2 _____
- XMPP01.3   33.0422 84.04A2 0A.0522 14.05A2 _____

```

Figure 56. *Select Processor / Control Unit (1)*

```

Select Processor / CU      Row 3 of 5 More: < >
Command ==> _____ Scroll ==> CSR

Select processors to change CU/processor parameters, then press Enter.

Control unit number . . : 0000      Control unit type . . . : 3422

      CU -----Unit Address . Unit Range + -----
/ Proc.CSSID Att ADD+ 1----- 2----- 3----- 4----- 5----- 6----- 7----- 8-----
- XMPP01.0   Y 00  00.256 _____
- XMPP01.1   Y 00  00.256 _____
- XMPP01.2   Y 00  00.256 _____
- XMPP01.3   Y 00  00.256 _____

```

Figure 57. *Select Processor / Control Unit (2)*

processor.

2. Select a processor and the *Select (connect/change)* action from the context menu (or action code **S**).

When a control unit is attached to multiple processors, you can use the *Group connect* action from the context menu (or action code **g**). This group action is particularly useful when performing symmetric changes, for example, on CPCs

defined in a z Systems cluster. The changes are applied to all selected processors, when you issued the change action against a group of processors. When you issue a change or group connect action, the following panel for processor-dependent control unit information is displayed:

Add Control Unit

Specify or revise the following values.

Control unit number . . . : 0099 Type : 2105
 Processor ID : FR38LPAR Raised floor production
 Channel Subsystem ID . . . :

Channel path IDs	07	08	*	*	*	_	_	_	+
Link address	80	81	_	_	_	_	_	_	+
Unit address	_	_	_	_	_	_	_	_	+
Number of units	_	_	_	_	_	_	_	_	

Logical address _ + (same as CUADD)

Protocol _ + (D,S or S4)
 I/O concurrency level . 2 + (1, 2 or 3)

3. On the Add Control Unit panel specify the channel paths that connect the control unit to the processor.
 If the control unit is attached to a switch, you have to define a link address for each channel path. The link address is the port to which the control unit attaches. If the control unit attaches only to one port, the link address is the same for each channel. For addressing the target control unit in a fabric containing cascade switching, a two-byte link address is used, which specifies as first byte the switch address and as second byte the port address to which the control unit is attached.

For a description what the link address is, see Figure 72 on page 164, Figure 73 on page 164 and Figure 75 on page 165.

Note: For managed control units, i.e., control units that can have managed channel paths assigned by DCM, you must indicate how many managed channel paths can be connected to the control unit. Enter at least one static channel path and the corresponding link address, and, in addition, an * (instead of the channel path ID and link address) for each managed channel path.

You must also specify the unit address and the number of units, that is the unit address range of I/O devices that the control unit recognizes. Serial control units may have specified only one unit address range starting with 00.

If the path to the control unit is not unique, and more than one serial control unit connects to the same channel path via the same link address, you have to specify a logical address (CUADD parameter). For more information refer to the explanation of the CUADD in the *IOCP User's Guide* for your processor.

4. Press the Enter key. HCD displays the updated Select Processor/Control Unit panel. There you may scroll to the right (using F20=Right) to see the data that you have entered on the previous panel.
5. Repeat defining processor attachment data for all processors the control unit should be attached to.
6. Press the Enter key to return to the Control Unit List.

Upgrading to two-byte link addresses

In a FICON fabric, all one-byte link addresses on a channel path may need to be migrated to a two-byte link address. HCD supports this definition change via the Change Channel Path Link Addresses dialog. If you specify a two-byte link address on a control unit for a specific channel path, and there are already one-byte link addresses specified on that path, the panel shown in Figure 58 appears.

This panel shows all link addresses specified for the specific channel path that must be changed. If there has been an entry switch defined for the channel, its ID and switch address (if defined) are displayed; else the displayed information is taken from the dynamic switch.

You can change the switch address. HCD then uses the modified value to preset the two-byte link addresses in the displayed New column. Its value is stored, if the entry switch is defined. Alternatively, the new two-byte link addresses can be entered.

Pressing Enter will change the control unit link addresses on the corresponding control units.

```
Change Channel Path Link Addresses

Command ==>> _____ Scroll ==>> PAGE

Specify or revise the following values.

Processor ID . . . . . : MCSS01
Channel Path ID . . . . : 81
Entry Switch ID . . . . : B0

Switch Address . . . . . 26

Control Unit -- Link Address --
Number CSSID Current New
6000 0 02 2602
6200 1 03 2603
6400 1 04 2604
6600 1 05 2605
7000 2 09 2609
0E06 2 FE 26FE
***** Bottom of data *****

F1=Help      F2=Split    F3=Exit     F5=Reset    F7=Backward
F8=Forward   F9=Swap     F12=Cancel  F22=Command
```

Figure 58. Change Channel Path Link Addresses

You may want to change link addresses from two byte to one byte. This is possible, as long as all affected control units are attached to the channel path switch. If a defined two-byte link address is changed to a one-byte link address on a given channel path, all other two-byte link addresses defined for control units attached to that channel have to be changed to a one-byte link address also.

Changing control units

To change control unit data, follow the same panel flow as for defining control units.

- Changing Control Unit Characteristics
- Changing Processor Attachment Data

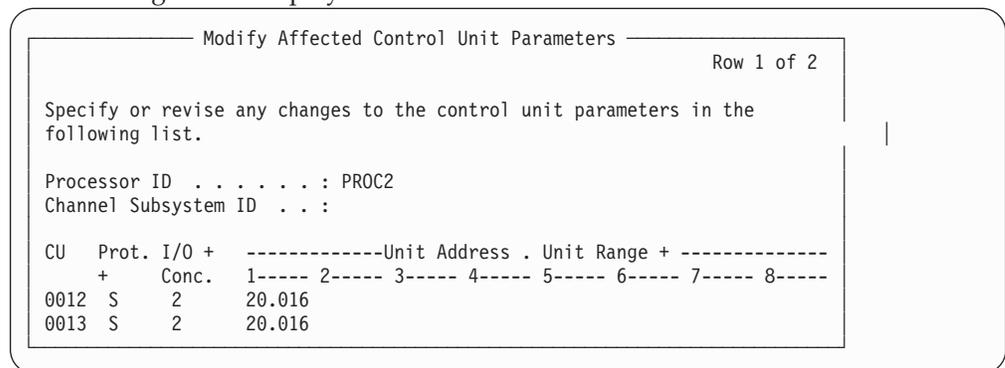
The following steps describe the panel flow and where you can change the data.

1. On the Control Unit List select a control unit and the *Change* action from the context menu (or action code **c**).
2. On the following Change Control Unit Definition panel you can change the following data:
 - Control unit number
 - Type-model
 - Serial number
 - Description
 - Connections to switches/ports

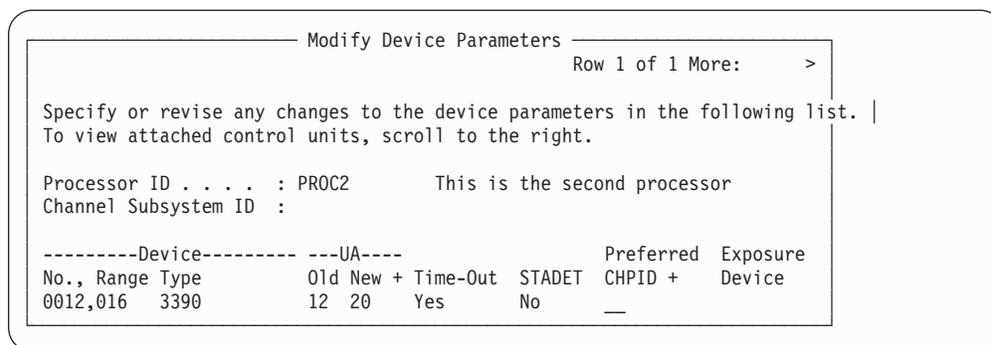
Note: You can also change these control unit definitions (except the control unit number and the connections to switches/ports) by simply typing over the appropriate columns on the Control Unit List.

3. After pressing the Enter key you see the Select Processor/Control Unit panel. Select a processor and the *Select (connect/change)* action from the context menu (or action code **s**).
4. On the following Change Control Unit Definition panel you can change the processor attachment data:
 - Channel paths / Link addresses
 - Unit addresses / Number of units
 - Logical address
 - Protocol
 - I/O concurrency level

When changing control unit data of control units that affect other control unit or device data (like unit address/ranges), a list is displayed that shows all affected control units and proposed new address ranges for those control units. A panel like the following one is displayed:



After you modified control unit data (like protocol, I/O concurrency level, or unit address range), and pressed the Enter key, the Modify Device Parameters panel is shown with the devices attached to the affected control units. The devices are grouped by ranges:



HCD proposes starting unit addresses for the listed device groups.

Use the F20=Right key to scroll to the right to see the attached control units. Accept or change the definitions for unit address (UA New), Time-Out, STADDET, and preferred CHPID.

Changing control unit attachment parameters for multiple processors

You can change control unit (CU) attachment parameters or attach a control unit for a group of processors. If all parameters to be changed are identical, use the following group action.

1. On the Control Unit List, select a control unit that is attached to the group of processors and use the *Change* action from the context menu (or action code **c**).
2. On the Change Control Unit panel press the Enter key. HCD then displays a Select Processor/Control Unit panel with a list of processors already defined (see Figure 56 on page 128).
3. Select the processors for which you want to change the control unit-processor definitions and use the *Group change* action from the context menu (or action code **g**).
4. The Change Control Unit Definition panel is displayed showing the values/attributes for the first processor in the group. An * in the Processor ID field indicates that you are using the *Group connect* action from the context menu and the changes will be applied to more than one processor.

Disconnecting control units from a processor

1. On the Control Unit List, select a control unit and the *Change* action from the context menu (or action code **c**). HCD displays the Change Control Unit Definition panel.
2. On the Change Control Unit Definition panel, press the Enter key. HCD displays the Select Processor/Control Unit panel.
3. On the Select Processor/Control Unit panel select a processor and the *Disconnect* action from the context menu (or action code **n**).

Disconnecting multiple control units from a processor:

If you want to disconnect multiple control units from one processor in one step, open the Control Unit list via the Channel Path List. On the Control Unit List, select one or multiple control units and use the *Disconnect* action from the context menu (or action code **n**).

Priming control unit data

You can prime your I/O configuration in a work IODF with the control unit serial number for the active processor. For the prerequisites for this function refer to "Prerequisites" on page 9.

To prime the control unit serial number, select the action *Prime serial number* from the context menu (or action code **i**) on the Control Unit List. The Confirm Priming Control Unit Data List shows the selected control units with the sensed data for the control unit types and serial numbers, and their corresponding definitions in the IODF.

```

----- Confirm Priming Control Unit Data List -----
                                                    Row 1 of 9
Command ==> _____ Scroll ==> CSR

Press Enter to confirm priming, or Cancel to leave the
list. A blank value will not change the IODF definition.

CU      ----- Type ----- --- Serial Number ---
Number actual      defined      sensed      defined
90C1          3990          33333      1
9100  3990-L03    3990      30984      2
9101  3990-L03    3990      30984      30984
9140  3990-L03    3990      37160      A7199      3
9141  3990-L03    3990      37160      A7199
9180  3990-L03    3990      37201
9181  3990-L03    3990      37201
91C0  3990-L03    3990      37163      37163
91C1  3990-L03    3990      37163      37163
***** Bottom of data *****

F1=Help      F2=Split      F3=Exit      F5=Reset
F7=Backward  F8=Forward    F9=Swap      F12=Cancel

```

- 1** The control unit serial number is defined in the IODF, but no sensed data is available on the active system.
- 2** No control unit serial number is defined in the IODF, but the sensed data of the active system is available. To confirm the sensed data, and to define them in the IODF, press Enter.
- 3** The control unit serial numbers that are defined in the IODF, and that are sensed are different. Press Enter, to overwrite the defined data by the sensed data.

Note: The sensed values can only be blanked out or left unchanged. Blank out the sensed values, if you don't want to change the defined IODF values. To confirm priming, press Enter. Use the F12=Cancel key, if you don't want to use the sensed values, and to leave the list.

Deleting control units

You can delete the definition of a control unit using the *Delete* action from the context menu (or action code **d**). Deleting a control unit means that all connections to channel paths, switches, and I/O devices are also deleted; these objects are not deleted.

Working with devices

Operating systems need I/O device data to address the devices. The CSS also needs the data to provide the required information for performing I/O operations to a specific device.

Defining devices

You need three steps to define an I/O device:

- Define device characteristics and control unit connection
- Define CSS-related definitions for a device
- Define OS-related definitions for a device (including EDT and esoteric group assignment - MVS-type only).

Before you define a device that should be defined to an operating system and to the channel subsystem (CSS), you must have defined the operating system configuration, processor, channel path, and control unit. HCD omits some steps if data is missing. For example:

- You cannot define the processor data for the device if the device is not attached to a control unit or the control unit is not attached to a processor.
- You cannot define the EDT/esoteric group data for the device until you have defined an EDT for the OS.

Defining device data

1. On the *Primary Task Selection* panel, select *Define, modify, or view configuration data* and on the resulting panel, select *I/O devices*. HCD displays an initial I/O Device List where devices with consecutive device numbers having the same definitions are automatically grouped together (Figure 59 on page 135).

A device group is shown as device number, range. A range value of one (1) is not explicitly shown. For example, the entry '0002,4 3390A' indicates a device group of four devices of type 3390A with consecutive device numbers from 0002 through 0005. Using action *Work with single I/O devices* from the context menu (or action code **S**) displays the I/O Device list showing all single devices defined in the IODF, with all device groups resolved (Figure 60 on page 135).

```

Goto Filter Backup Query Help
-----
I/O Device List          Row 1 of 2370 More:  >
Command ==> _____ Scroll ==> CSR

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

-----Device----- --#--- -----Control Unit Numbers + -----
/ Number  Type +      CSS OS 1--- 2--- 3--- 4--- 5--- 6--- 7--- 8---
- 0000    3380          1
- 0001    3390A      3  1 0001 0002 _____
- 0001    3390A      2  1 0001 _____
- 0001    3390A      1
- s0002,4 3390A      2  1 0001 _____
- 0002,8 3390A      3  1 0001 0002 _____
- 0006,3 3390      2  1 0001 _____
- 000A    3380          2  1 0001 _____
- 000B    3820          1
- 000C,12 3590          1
# FFFC    CFS          FFFE _____
# FFFD    CFS          FFFE _____
# FFFF    CFS          FFFE _____
# FFFF    CFS          FFFE _____

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

```

Figure 59. I/O Device List with device groups

The # sign in front of a row indicates that this row is disabled. You cannot modify or delete it. In the example from Figure 59, you can see four devices of type CFS that are used for coupling facility connections.

If you scroll to the right in the I/O Device List, you can see additional columns *PU* (showing the PPRC usage), *Serial-#*, *Description*, and *VOLSER*.

```

Goto Filter Backup Query Help
-----
I/O Device List          Row 9 of 49852 More:  >
Command ==> _____ Scroll ==> CSR

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

-----Device----- --#--- -----Control Unit Numbers + -----
/ Number  Type +      CSS OS 1--- 2--- 3--- 4--- 5--- 6--- 7--- 8---
- 0002    3390A      2  1 0001 _____
- 0002    3390A      3  1 0001 0002 _____
- 0003    3390A      2  1 0001 _____
- 0003    3390A      3  1 0001 0002 _____
- 0004    3390A      2  1 0001 _____
- 0004    3390A      3  1 0001 0002 _____
- 0005    3390A      2  1 0001 _____
- 0005    3390A      3  1 0001 0002 _____
- 0006    3390      2  1 0001 _____
- 0006    3390A      3  1 0001 0002 _____
- 0007    3390      2  1 0001 _____
- 0007    3390A      3  1 0001 0002 _____

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

```

Figure 60. I/O Device List with single devices

Columns CSS and OS state the number of channel subsystems and operating systems accessing the device. If the I/O Device List is called from the Processor List or Channel Subsystem List, the number in the *IM* column states how many partitions (images) of the selected processor or channel subsystem are accessing the device. For basic processors this value is one.

- If the I/O Device List is called from either
- the Operating System Configuration List
 - the Processor List for SMP processor
 - the Channel Subsystem List for XMP processor

using action *Work with attached devices* from the context menu (or action code **u**), then the list contains an additional column *SS* which indicates, if applicable, in which subchannel set the device should be placed.

2. Use F11=Add to add I/O devices. The data-entry fields are shown in the following figure, with sample data:

Add Device

Specify or revise the following values.

Device number 01E1 + (0000 - FFFF)
 Number of devices 8
 Device type 3390A +

Serial number _____
 Description PAV alias device _____

Volume serial number _____ (for DASD)
 PPRC usage _ + (for DASD)

Connected to CUs . . 01E1 _____ +

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

Figure 61. Add Device

In the *Device number* field, you can use the F4=Prompt key to have a list displayed containing unused device number ranges. If you select a proposal from this list, HCD fills *Device number* and *Number of devices* with the selected values.

The **Add Device** panel can also be used to specify the control units the devices are connected to.

For a DASD device, you can optionally define its peer-to-peer remote copy (PPRC) usage type as either:

- Duplex (D)
- Flashcopy (F)
- Simplex (S)
- Utility (U)
- Nonsysplex (N)

Whereas a classification as *Flashcopy*, *Simplex*, *Utility*, or *Nonsysplex* is only of a descriptive character, the *Duplex* usage type of a DASD device triggers the following: Duplex devices, attached to a primary operating system configuration with OFFLINE=YES are defined to an optional D/R site OS configuration with OFFLINE=NO and vice versa, when the D/R site OS configuration is generated (see also “D/R site OS configurations” on page 78).

Since VM dummy devices are definable with an arbitrary device type, a device with an unknown device type is accepted by HCD. It is treated like an unsupported device with the device type DUMMY. For MVS-type systems, you have to explicitly define the device as DUMMY.

Defining multiple devices in one step: You can define, in one operation, a group of I/O devices of the same type and with consecutive device numbers. You define the group by specifying the first device number and the number of devices in the group. Then HCD applies the definition to all devices in the group. On the I/O Device List, you can type over the values that should be different.

Use and definition of serial number of device: HCD allows you to assign the same device number to more than one I/O device; that is, device numbers alone do *not* uniquely identify a device in an IODF. To clearly identify devices, HCD keeps track of each occurrence of the same device number by appending an internal suffix to the device number.

When *activating a configuration dynamically*, HCD might be unable to determine whether certain I/O devices in the *currently active IODF* and the *IODF to be activated* are physically the same. This may happen, if the new IODF was not created by copying or updating the current IODF but was newly created by migrating with IOCP or using the HCD dialog. In this case HCD is unable to determine which of the devices are physically identical.

To avoid problems when activating a configuration dynamically, you should check if more than one device uses the same device number attached to the same control units in the current IODF and in the newly created (not copied) IODF. If so, specify the same *serial number* for the devices that HCD should treat as physically the same.

Defining CSS-related definitions for a device

If you have defined a connection to a control unit on the *Add Device panel*, and the control unit is connected to a processor, then HCD displays the *Device / Processor Definition* panel (Figure 62) that shows the processors to which the control units are attached.

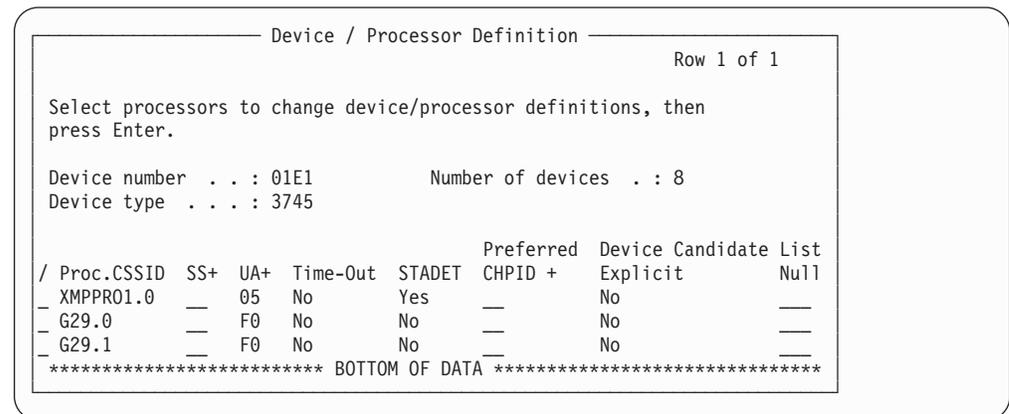


Figure 62. Device / Processor Definition

On the *Device / Processor Definition* panel you can proceed in two ways:

- You can specify the CSS-related definitions directly by typing over the fields in each column. If you want to specify an explicit device candidate list for a device, type 'yes' into column *Device Candidate List - Explicit*. This leads you to panel *Define Device Candidate List* (Figure 64 on page 139).
- You can select a processor and press the Enter key. The *Define Device / Processor* panel is displayed (Figure 63 on page 138). From this panel you can edit the same values as shown in the *Device / Processor Definition* panel.

```

Define Device / Processor

Specify or revise the following values.

Device number . . . . : 01E1          Number of devices . . . . : 8
Device type . . . . . : 3745
Processor ID . . . . . : XMPPR01
Channel subsystem ID : 0

Subchannel set ID . . . . . _ +
Unit address . . . . . _E1 + (Only necessary when different from
                             the last 2 digits of device number)

Time-Out . . . . . No (Yes or No)
STADET . . . . . Yes (Yes or No)

Preferred CHPID . . . . . _ +
Explicit device candidate list . No (Yes or No)

```

Figure 63. Define Device / Processor

Defining the subchannel set for a device: Starting with z9 EC processors, each channel subsystem contains more than one subchannel set (SS 0, SS 1), where you can place the devices. Starting with z/OS V1R7 HCD, you can place PAV alias devices (types 3380A and 3390A) into an alternative subchannel set. In SS 0, you can place 63.75K devices, and in SS 1 you can place 64K-1 PAV alias devices.

Starting with zEnterprise processors, each channel subsystem contains a third subchannel set (SS 2). Starting with z/OS V1R10 HCD, you can place PAV alias devices (types 3380A and 3390A), PPRC secondary devices (type 3390D) and DB2 data backup volumes (type 3390S) into an alternate subchannel set. You cannot define 3390D and 3390S devices in subchannel set SS 0.

You can specify the subchannel set ID for a device either in column **SS** of Figure 62 on page 137 or in field **Subchannel set ID** of Figure 63.

HCD messages that refer to a device in a subchannel set with a subchannel set ID > 0 will display the device number in the format *n-devnumber* where n is the subchannel set ID. For example, the device 1234 located in subchannel set 1 will show up as *1-1234*. A device 4567 in subchannel set 0 will further on be shown as *4567*.

Rules for placing devices into subchannel sets:

Observe the following rules and recommendations when working with different subchannel sets:

- There is no required correspondence between device numbers in the subchannel sets. For example,
 - devices in the range 8000-807F in SS0
 - devices in the range 8000-807F in SS1 (PAV alias devices)

may relate to completely separate devices. However, you can use this feature to have PAV base and aliases in different subchannel sets, but with the same device numbers.

- Unit addresses of base and alias devices on a single control unit must be unique. These cannot be duplicated across subchannel sets. So if you want to define the PAV base and alias devices in the range 8000-807F in different subchannel sets, but on the same control unit, you can define them like follows:

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

- You can use dynamic reconfiguration to move eligible devices from SS 0 to an alternate subchannel set.

Restricting Partition Access for Devices: You can restrict logical partition access to an I/O device on a shared channel path by using the explicit device candidate list to select which logical partitions can access that I/O device. On the *Define Device / Processor* panel enter Yes or No in the Explicit device candidate list field to specify whether you want to restrict logical partition access to an I/O device:

- A No specifies that all logical partitions can access this I/O device. No is the default; all logical partitions are in this I/O device's candidate list.
- A Yes specifies that only your selected logical partitions can access this I/O device. Note that the partition must also be in the channel path access or candidate list to access the device. On the Define Device Candidate List, place a slash (/) character to the left of each selected Partition Name.

If you specify Yes in the Explicit device candidate list field, the following panel is displayed, showing possible candidate partitions:

Define Device Candidate List Row 1 of 6

Select one or more partitions to allow them to access the device, or ENTER to continue without selection.

Device number . . . : 01E1 Number of devices . . . : 8
 Device type : 3745
 Processor ID : PROC1 This is the main processor
 Channel subsystem ID :

/ Partition Name	Description	Reachable
- PROD1	First production partition	Yes
- PROD2	Second production partition	Yes
- TEST1	First test system	No
- TEST2	Second test system	Yes
- TEST3	CF partition	No
- TEST4	OS partition	No

Figure 64. Define Device Candidate List

A Yes in the Reachable column indicates that the device can be reached from the respective partition, through at least one physical channel. You can only include reachable partitions into the explicit device candidate list by typing a slash (/) into the action column. Deleting the slash means to remove the respective partition from the device candidate list.

Null device candidate list 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 empty (or null) device candidate list for one or more CSSs. You create a null device candidate list for a device either by deselecting all candidate partitions from an existing list or by not selecting any partition for a new list.

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. If you define an explicit device candidate list for a device, the *Device / Processor Definition* panel (Figure 62 on page 137) indicates whether this candidate list is a null device candidate list in

column *Device Candidate List - Null*. If no partition is allowed to have access to the device, value Yes is shown, otherwise value No. This field is left blank if no explicit device candidate list exists for the selected device (which is the default when creating new devices).

Defining OS-related definitions for a device

1. After pressing the Enter key on the *Define Device / Processor* panel, the *Device / Processor Definition* panel is displayed again. Select another processor or press the Enter key again to display the *Define Device to Operating System Configuration* panel that shows all the defined OS configurations.

```

Define Device to Operating System Configuration
Row 1 of 2

Select OSs to connect or disconnect devices, then press Enter.

Device number . . : 01E1          Number of devices : 8
Device type . . . : 3390A

/ Config. ID  Type  SS Description          Defined
_ OPSYS01    MVS   MVS or z/OS operating system
_ OPSYS02    VM    VM operating system
*****
***** BOTTOM OF DATA *****

```

Select an operating system and the *Select (connect/change)* action from the context menu (or action code **S**).

As described in “Defining the subchannel set for a device” on page 138, starting with 2094 (z9 EC) processors, you can place PAV alias devices (types 3380A and 3390A) into SS 1.

If you define a PAV alias device, as shown in our example from Figure 61 on page 136, HCD displays the *Specify Subchannel Set ID* panel that asks for the subchannel set where you want to place the device. The default depends on the value given for the corresponding CSS definition.

```

Specify Subchannel Set ID

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

Configuration ID . . : OPSYS01
Device number . . . : 01E1          Number of devices : 8
Device type . . . . : 3390A

Subchannel Set ID   1 +

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

```

Figure 65. Specify Subchannel Set ID

2. Pressing Enter on the dialog from Figure 65 brings you to the following panel where you can now define the data about device parameters and features that are required by the operating system configuration.

```

Define Device Parameters / Features
Row 1 of 4

Specify or revise the values below.

Configuration ID . . : OPSYS01      MVS or z/OS operating system
Device number . . . : 01E1         Number of devices : 8
Device type . . . . : 3390A

Parameter/
Feature Value +      R Description
WLMPAV   Yes         Device supports work load manager
***** BOTTOM OF DATA *****

```

The Parameter/Feature fields vary depending on the I/O device type and operating system type.

A plus sign (+) in the Value column indicates that you may use F4=Prompt to get a list of possible values for the parameter/feature in the same row. Note that not all parameters are promptable.

A Y in the R column indicates that a value for the parameter/feature in the same row is required.

You accomplish the change by accepting the default values or by changing the Value entries and pressing the Enter key. The default values are set in the UIM for the device type. For parameters you can specify different default values via the OS_PARM_DEFAULT keyword in the HCD profile.

- For eligible devices, after you have defined the device parameter and feature data and pressed the Enter key, HCD displays the *Assign/Unassign Device to Esoteric* panel.

```

Assign/Unassign Device to Esoteric
Row 1 of 2

Specify Yes to assign or No to unassign. To view devices already
assigned to esoteric, select and press Enter.

Configuration ID : OPSYS01      MVS or z/OS operating system
Device number . . : 01E1         Number of devices : 8
Device type . . . : 3390A         Generic . . . . . : 3390A

/ EDT.Esoteric Assigned Starting Number Number of Devices
_ A1.ES001      No          _____
_ A2.ES002      No          _____
***** BOTTOM OF DATA *****

```

- On the Assign/Unassign Devices to Esoterics panel, overwrite the values in the Assigned column to assign (Yes) or unassign (No) devices to the selected esoterics.

If you do not want to assign a complete group of devices, you can limit the range by specifying a starting number and the number of devices. If you omit the number of devices, 1 is assumed.

Changing devices

To change device data, you have to follow the same panel flow as for defining a device:

- Changing device and control unit definitions

2. Changing CSS-related definitions
3. Changing OS-related definitions

The following steps describe the panel flow and where you can change which data.

1. On the I/O Device List, select a device or a group of devices and the *Change* action from the context menu (or action code **c**). HCD shows the following panel:

Change Device Definition

CBDP30

Specify or revise the following values.

Device number : 0005 (0000 - FFFF)
 Number of devices : 4
 Device type : 3390

Serial number _____ +
 Description _____

Volume serial number _____ + (for DASD)

PPRC usage _ + (for DASD)

Connected to CUs . 0001 _____ +

ENTER to continue.

Depending on whether you invoke this action for a single device or a group of devices, the line *Number of devices* shows how many devices are affected by the change.

2. On the *Change Device Definition* panel you can change device and control unit definitions such as:
 - Serial number
 - Description
 - Volume serial number
 - PPRC usage
 - Control unit connections
3. After pressing the Enter key, the *Device / Processor Definition* panel is displayed. Select a processor and press the Enter key to change the following CSS-related definitions:
 - Subchannel set ID
 - Unit address
 - Time-Out
 - STADET
 - Preferred CHPID
 - Explicit device candidate list
4. After pressing the Enter key twice, the Define Device to Operating System Configuration panel is displayed. Select an operating system and the *Select (connect/change)* action from the context menu (or action code **s**) if you want to change the following OS-related definitions:
 - Parameters/Features
 - Assignments to esoterics

5. After pressing the Enter key again, the Assign/Unassign Device to Esoteric panel is displayed. If you want to change the assignment of devices to esoterics, type over the values in the Assigned column by either Yes or No.
6. Press the Enter key twice to return to the I/O Device List.

Changing CSS-related definitions of a group of devices

You can change CSS-related definitions of a group of devices using the *CSS group change* action. This helps you, for example, to attach a group of DASDs to another control unit. To do this, the devices to be changed must be in the same device group, that is, they must all be of type, for example, DASD or TAPE.

1. On the *I/O Device List* select one or more devices and use the *CSS group change* action from the context menu (or action code **g**). The Change Device Group panel is displayed.

HCD displays the definition of the first device in the group. You can modify this definition and HCD applies the definition to all devices in the group.

2. After pressing the Enter key, HCD displays the Change Device Group / Processor Definition panel, where you can select the processors for which you want to change the CSS-related definitions. For an example of this panel, see “Defining CSS-related definitions for a device” on page 137.

Changing esoterics for a group of devices

For a description of how to change esoterics for multiple devices, refer to “Adding devices to esoterics” on page 84.

Changing OS-related definitions of a group of devices

You can change OS-related definitions for a group of devices using the *OS group change* action (or action code **o**). This helps you, for example, to attach a group of devices to another operating system. The device parameter/features will be the same for all devices in the group.

If you want to change OS-related definitions for PAV devices, HCD displays a similar dialog as shown in Figure 65 on page 140 which lets you change or specify the subchannel set ID where to place the device or the device group.

1. On the I/O Device List for device groups or single devices select one or more devices or groups and use the *OS group change* action from the context menu (or action code **o**). HCD displays the Change Device Group / Operating System Configuration dialog.

```

Change Device Group / Operating System Configuration
Row 1 of 1
Command ==> _____ Scroll ==> PAGE

Select OSs to connect or disconnect devices, then press Enter.

/ Config. ID  OS Type  Description                Defined
_ OPSYS01    MVS       MVS or z/OS operating system  Yes
***** Bottom of data *****

F1=Help      F2=Split    F3=Exit      F4=Prompt    F5=Reset
F7=Backward  F8=Forward  F9=Swap      F12=Cancel   F22=Command

```

HCD applies the OS related changes to all selected devices, if a user performs an explicit action (for example, *Select (connect/change)* or *Disconnect from OS*, see next step). You might need to also disconnect from those operating systems, which are not shown as connected (status is shown only for the first device) to ensure, that all selected devices are disconnected.

2. If you want to disconnect the selected device group(s) from specific operating systems, select those operating systems and action *Disconnect from OS* (action code **n**) from the context menu.

Otherwise, select the operating system to which you want to attach the group of devices and the *Select (connect/change)* action from the context menu (or action code **s**). HCD displays the following dialog:

```

Define Device Group Parameters / Features
Row 1 of 4
Command ==> _____ Scroll ==> PAGE

Specify or revise the values below.

Configuration ID . : OPSYS01      MVS or z/OS operating System

Parameter/
Feature  Value  P Req.  Description
ADAPTER  TYPE7  + Yes   Channel adapter type
OFFLINE  No     +       Device considered online or offline at IPL
DYNAMIC  Yes    +       Device supports dynamic configuration
OWNER    VTAM   +       Subsystem or access method using the device
***** Bottom of data *****

F1=Help      F2=Split    F3=Exit      F4=Prompt    F5=Reset
F7=Backward  F8=Forward  F9=Swap      F12=Cancel   F22=Command

```

3. You accomplish the change by accepting the default values or by changing the Value entries and pressing the Enter key.

The specified device parameters/features are applied to all devices of the group.

Changing the DYNAMIC, LOCANY or OFFLINE parameter of a group of devices

You can change the DYNAMIC, LOCANY or OFFLINE parameter of a group of devices using the *Attribute group change* action. This function helps you to change parameters for a group of devices without having to use the *Change* action for each device individually.

This function can only be invoked from the I/O Device List accessible from the Operating System Configuration List.

1. On the *Primary Task Selection* panel, select *Define, modify, or view configuration data* and on the resulting panel the object *Operating system configurations*.
2. Select an operating system and select the *Work with attached Devices* action from the context menu (or action code **u**). HCD displays the I/O Device List.
3. Select one or more devices on the I/O Device List and the *Attribute Group change* action from the context menu (or action code **e**). HCD displays the Attribute Group Change panel:

Attribute Group Change

For all devices in the selected group, choose whether ...

- 1. Allow dynamic configuration DYNAMIC=YES
- 2. Do not allow dynamic configuration ... DYNAMIC=NO
- 3. UCB can reside in 31 bit storage LOCANY=YES
- 4. UCB can not reside in 31 bit storage .. LOCANY=NO
- 5. Device is set offline at IPL OFFLINE=YES
- 6. Device is set online at IPL OFFLINE=NO

Select the appropriate parameter.

HCD only changes the single parameter for all devices of the group, leaving the other parameters/features of the group unchanged.

Changing type/model of a group of devices

You can change the type or model for a group of devices using the *Device type group change* action. However, you have to make sure that all devices to be changed in one step have the same device type and model. The control units the devices are attached to, have to support the attachment of the new device type as well, and required parameters have to be identical. The new device type has to be supported by the same operating system type.

1. Select one or more devices on the I/O Device List.
2. Use the *Device type group change* action from the context menu (or action code **t**). HCD displays the Device Type Group Change panel.

Device Type Group Change

Specify a new device type-model.

Current device type-model . : 3380

New device type-model 3390 _____ +

Specify a new device type-model.

Changing the subchannel set placement for a group of devices

You can change the placement of PAV alias devices any time, for example, if you want to migrate PAV alias devices into a subchannel set of a new processor. From the I/O Device List showing device groups or single devices, use action *Subchannel Set ID group change* from the context menu (or action code **m**). HCD displays the following dialog where you can specify the new ID of the subchannel set.

Note: When defining or changing the subchannel set placement for devices, you need to observe certain rules. For more information, read “Defining CSS-related definitions for a device” on page 137 and refer to the *z/OS HCD Planning*.

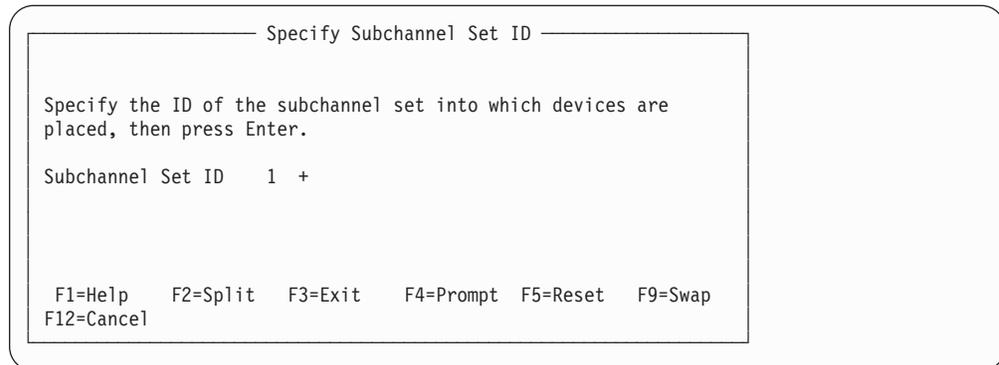


Figure 66. Specify Subchannel Set ID

If at least one of the selected devices has a connection defined to a processor supporting multiple subchannel sets, HCD displays a dialog where you can select from the eligible channel subsystems where to move the devices.

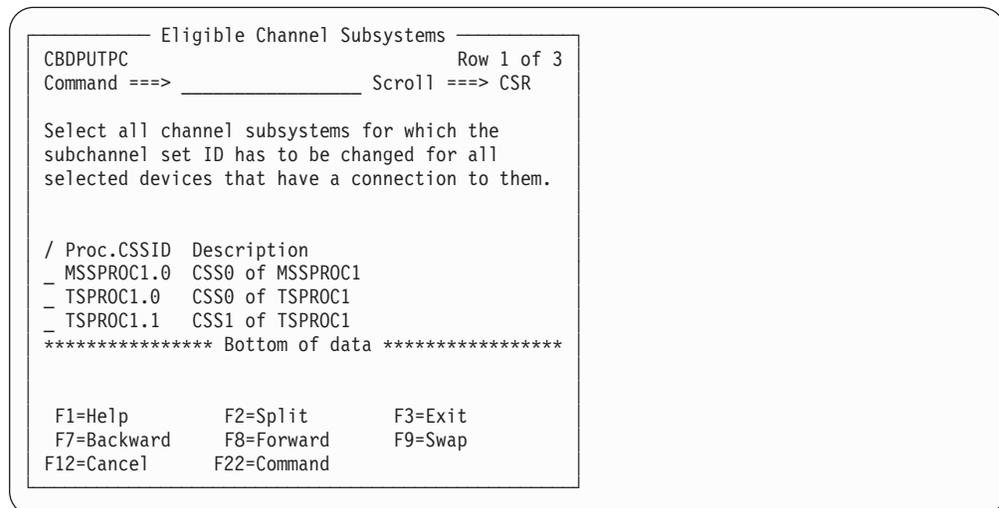


Figure 67. Eligible Channel Subsystems

Also, if at least one of the selected devices has a connection defined to an operating system configuration, HCD displays a dialog listing all OS configurations that have connections to any of the selected devices. You can select all OS configurations for which you want to change the subchannel set ID for the selected devices.

```

----- Eligible Operating System Configurations -----
CBDPUTOC                                     Row 1 of 1
Command ==> _____ Scroll ==> CSR

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

/ Config.ID   Type           Description
- ZOS17       MVS           first z/OS 1.7 operating system
- Z17SCND     MVS           second z/OS 1.7 operating system
***** Bottom of data *****

F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward
F9=Swap      F12=Cancel   F22=Command

```

Figure 68. Eligible Operating System Configurations

Changing the device number

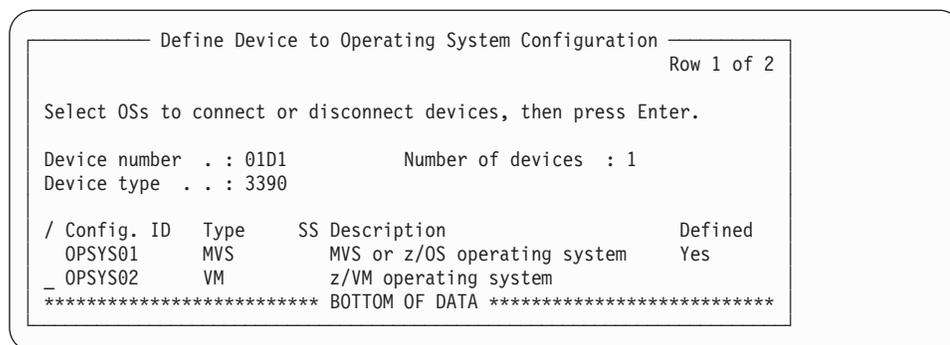
To change the number of a device:

1. Remove the connections to the control units for the devices to be changed as follows:
 - a. On the I/O Device List, select the devices to be changed and the *CSS group change* action from the context menu (or action code **g**). The Change Device Group panel is displayed.
 - b. Remove the control unit numbers from the panel and press the Enter key.
2. On the I/O Device List, select the device and the *Add like* action from the context menu (or action code **a**). The Add Device panel is displayed.
3. Specify the new number for the device and the control unit numbers to which the devices are to be attached. Press the Enter key. HCD now displays a series of panels showing the settings of the previously selected device (the one to be changed). The settings are propagated to the new devices. Press the Enter key until HCD redisplay the I/O Device List now showing the new device.
4. Delete the old device by selecting the device and selecting the *Delete* action from the context menu (or action code **d**).

Disconnecting devices from an operating system

Perform the following steps to disconnect a device from an operating system.

1. On the I/O Device List select a device and the *Change* action from the context menu (or action code **c**).
2. On the *Change Device Definition* panel, press the Enter key.
3. On the *Device / Processor Definition* panel, press the Enter key once again. HCD displays the *Define Device to Operating System Configuration* panel.



- On the Define Device to Operating System Configuration panel select an operating system and the *Disconnect from OS* action from the context menu (or action code **n**). The Define Device to Operating System Configuration panel is displayed again without showing a Yes in the Defined column.

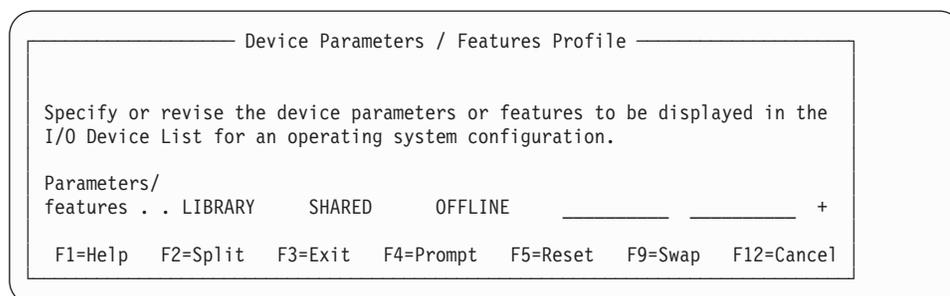
Disconnecting multiple devices from an operating system:

If you want to disconnect multiple devices from one operating system in one step, open the I/O device list via the OS configuration list. On the I/O Device List, select one or multiple devices and use *Disconnect from OS* action from the context menu (or action code **n**).

Showing or hiding parameter/feature definitions of devices

You can define up to five parameters/features for a device that can be shown on the I/O Device List in addition to the default information. These parameters/features will be retained across sessions.

- On the Operating System Configuration List, use the *Work with attached devices* action from the context menu (or action code **u**).
- On the following I/O Device List, select the *Show parameters/features* pull-down choice from the *Show/Hide* action bar (no action code available).
- On the following Device Parameters/Features Profile, you can specify up to five parameters/features that will be displayed on the I/O Device List. HCD saves your settings across sessions.



- On the I/O Device List, use the F20=Right key to scroll to the rightmost part of the panel, where the information is displayed. Note that the DYNAMIC parameter and LOCANY parameter are default information that is also shown on the leftmost part of the I/O Device List in columns **D** and **L**.

```

Goto Show/Hide Filter Backup Query Help
-----
I/O Device List          Row 1 of 11 More: <

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

Configuration ID . . . : OPSYS01      MVS or z/OS operating system

-----Device-----
/ Number  Type          LIBRARY  SHARED  OFFLINE
- 0001    3278-3
- 0098    9033
- 00C1    3480          No
- 01D1    3390
- 01D2    3390
- 01D3    3390
- 01D4    3390
- 01D5    3390
- 01D6    3390
- 01D7    3390
- 01D8    3390

```

You can filter the shown devices by device parameters and features using the *Set Filter* function. On the Filter I/O Device List, you can specify a value for any displayed parameter/feature you want to use for filtering.

In case you no longer need the parameters/features to be displayed, you use *Hide device parameters/features* pull-down choice from the *Show/Hide* action bar on the I/O Device List.

Priming device data

You can prime your I/O configuration in a work IODF with the device serial numbers and volume serial numbers (VOLSER) for the active processor. For the prerequisites for this function refer to “Prerequisites” on page 9.

To prime, select the action *Prime serial number and VOLSER* from the context menu (or action code **i**) on the I/O Device List.

The Confirm Priming Device Data List shows the selected devices with the sensed data for the device types and serial numbers, and their corresponding definitions in the IODF. For DASD devices, the sensed VOLSER is also shown on this panel.

```

----- Confirm Priming Device Data List -----
                                         Row 1 of 8
Command ==> _____ Scroll ==> CSR

Press Enter to confirm priming, or Cancel to leave the list. A blank value
will not change the IODF definition.

Device ----- Type ----- --- Serial Number --- --- VOLSER ---
Number actual   defined   sensed   defined   sensed defined
0AF0 3390-B3C   3390     B9888    B9888    PETHS2 PETHS2
0AF1 3390-B3C   3390     B9888    B9888    D83NE2 D83NE2
0AF2 3390-B3C   3390     B9888    B9888    D83WL4 D83WL4
0AF3 3390-B3C   3390     B9888    B9888    D83WL5 PETOFF 1
0AF4 3390-B3C   3390     B9916    A9999    PETCAT PETCAT
0AF5 3390-B3C   3390     B9916    B9916    PETDL2 PETDL2
0AF6 3390-B3C   3390     B9916    B9916    TSOPAK 2
0AF7 3390-B3C   3390     B9916    B9916    CMNAF7 CMNAF7
***** Bottom of data *****

F1=Help      F2=Split    F3=Exit     F5=Reset    F7=Backward
F8=Forward   F9=Swap     F12=Cancel

```

1 The values that are defined in the IODF and that are sensed are different. Press Enter, to overwrite the defined data by the sensed data.

2 No values are defined in the IODF, but the sensed data of the active system is available. To confirm the sensed data, and to define them in the IODF, press Enter.

Note: The sensed values can only be blanked out or left unchanged.

Blank out the sensed values, if you don't want to change the defined IODF values.

To confirm priming, press Enter.

Use the F12=Cancel key, if you don't want to use the sensed values, and to leave the list.

Deleting devices

You can delete the definition of a device or a device group using the *Delete* action from the context menu (or action code **d**). If you delete a device, all connections to the operating system including esoterics and EDTs are also deleted.

Working with operating system consoles

The following procedure describes how to specify which devices MVS can use as NIP consoles and which devices VM can use as VM consoles. Before you can define consoles you must have defined these I/O devices to the operating system.

1. On the *Primary Task Selection* panel, select *Define, modify, or view configuration data* and on the resulting panel the object *Operating system configurations*. HCD displays the Operating System Configuration List showing all OS configurations currently defined in the IODF.
2. Select an OS configuration and the *Work with consoles* action from the context menu (or action code **n**). HCD displays the NIP Console List or VM Console List (depending on the type of the selected operating system).

```

----- NIP Console List -----
Goto Backup Query Help
-----
Row 1 of 1

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

Configuration ID . : OPSYS01      MVS or z/OS operating system

/ Order  Device  Device Type
  Number  Number
_   1     0001   3278-3
***** BOTTOM OF DATA *****

```

3. Use F11=Add to define each console. The following panel is displayed:

```

----- Add NIP Console -----

Specify the following values.

Device number of console . . . . . ____

Order number . . . . . 1

```

The order number is the sequence the consoles are used by the operating system.

Changing operating system consoles

You can change the order number of an operating system console by just typing over the corresponding column or by using the *Change* action from the context menu (or action code **c**) on the Console List.

Deleting operating system consoles

You can delete the definition of an operating system console using the *Delete* action from the context menu (or action code **d**) on the Console List. The devices are not deleted.

Working on IODFs enabled for multi-user access

When multiple users concurrently use the same IODF, a user's changes are not immediately refreshed in the views of the other users. However, each user has a consistent view of the data either from the initial access to the IODF or after each last update that he had applied to the IODF.

In a few scenarios, this information unit demonstrates how HCD applies these rules when two users, *UserA* and *UserB* concurrently view or update an IODF:

- "Simultaneously updating and viewing an IODF"
- "Concurrently updating an IODF" on page 152
- "Immediately reflecting changes during concurrent updates" on page 153.

Simultaneously updating and viewing an IODF

Imagine that both users *UserA* and *UserB* invoke the *Channel Subsystem List* of the same IODF. *UserA* wants to delete channel subsystems CSS 1 and CSS 3, and *UserB* wants to work with the partitions of CSS 1.

UserA:					UserB:				
Channel Subsystem List					Channel Subsystem List				
Processor ID . . . : BBBX					Processor ID . . . : BBBX				
	CSS Devices in SS0		Devices in SS1			CSS Devices in SS0		Devices in SS1	
/ ID	Maximum + Actual		Maximum + Actual		/ ID	Maximum + Actual		Maximum + Actual	
0	65280 28		65535 0		0	65280 28		65535 0	
d 1	65280 28		65535 0		p 1	65280 28		65535 0	
2	65280 0		65535 0		2	65280 0		65535 0	
d 3	65280 0		65535 0		3	65280 0		65535 0	

Both users press Enter. While **UserA** sees the updated *Channel Subsystem List*, **UserB** sees the partitions of the meanwhile deleted CSS 1, because he still views the state of the IODF as loaded from storage. He will get a refreshed view after applying an update on the IODF.

UserA:					UserB:		
Channel Subsystem List					Partition List		
Processor ID . . . : BBBX					Processor ID . . . : BBBX		
CSS Devices in SS0					Configuration mode . : LPAR		
Devices in SS1					Channel Subsystem ID : 1		
/ ID	Maximum + Actual		Maximum + Actual		/ Partition Name	Number Usage	
0	65280 28		65535 0		PART11	1 CF/OS	
2	65280 0		65535 0		PART12	2 CF/OS	
					PART13	3 CF/OS	

Concurrently updating an IODF

Both users **UserA** and **UserB** start on the *Channel Subsystem List* of the same IODF. **UserA** invokes action *Change* on CSS 1 and **UserB** updates *Maximum Devices in SS0* for CSS2 and CSS3 from 65280 to 64512 and additionally invokes action *Delete* on the same CSS 1 maybe a few seconds later.

UserA:					UserB:				
Channel Subsystem List					Channel Subsystem List				
Processor ID . . . : BBBX					Processor ID . . . : BBBX				
	CSS Devices in SS0		Devices in SS1			CSS Devices in SS0		Devices in SS1	
/ ID	Maximum + Actual		Maximum + Actual		/ ID	Maximum + Actual		Maximum + Actual	
0	65280 28		65535 0		0	65280 28		65535 0	
c 1	65280 28		65535 0		d 1	65280 28		65535 0	
2	65280 0		65535 0		2	64512 0		65535 0	
3	65280 0		65535 0		3	64512 0		65535 0	

HCD displays panel *Change Channel Subsystem* for **UserA** while **UserB** receives message CBDA340I. When **UserB** returns from the *Message List* to the *Channel*

Subsystem List, the panel is not refreshed and *UserB*'s updates are kept on the screen. Thus, *UserB* can retry his update request several times until *UserA* releases the lock on the IODF.

UserA:	UserB:
Change Channel Subsystem Specify or revise the following values. Processor ID : BBBX Channel subsystem ID . . . : 1 Description _____ Maximum number of devices in subchannel set 0 . . 65280 + ...	Message List Messages are sorted by severity. Select one or more, then press Enter. / Sev Msg. ID Message Text _ E CBDA340I IODF USERA.IODF00.TEST.WORK # is currently being updated # by HCD user USERA on system # SCLM.

Immediately reflecting changes during concurrent updates

Both users *UserA* and *UserB* start on the *Channel Subsystem List* of the same IODF. *UserA* invokes action *Delete* on CSS 1 and CSS 2, locking the IODF, and *UserB* at the same time invokes action *Change* on several CSSs.

UserA:	UserB:																																																
Channel Subsystem List Processor ID : BBBX <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 10%;">CSS ID</th> <th style="width: 15%;">Devices in SS0</th> <th style="width: 15%;">Devices in SS1</th> </tr> <tr> <th></th> <th>/ ID</th> <th>Maximum + Actual</th> <th>Maximum + Actual</th> </tr> </thead> <tbody> <tr> <td></td> <td>0</td> <td>65280 28</td> <td>65535 0</td> </tr> <tr> <td>d</td> <td>1</td> <td>65280 28</td> <td>65535 0</td> </tr> <tr> <td>d</td> <td>2</td> <td>65280 0</td> <td>65535 0</td> </tr> <tr> <td></td> <td>3</td> <td>65280 0</td> <td>65535 0</td> </tr> </tbody> </table>		CSS ID	Devices in SS0	Devices in SS1		/ ID	Maximum + Actual	Maximum + Actual		0	65280 28	65535 0	d	1	65280 28	65535 0	d	2	65280 0	65535 0		3	65280 0	65535 0	Channel Subsystem List Processor ID : BBBX <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 10%;">CSS ID</th> <th style="width: 15%;">Devices in SS0</th> <th style="width: 15%;">Devices in SS1</th> </tr> <tr> <th></th> <th>/ ID</th> <th>Maximum + Actual</th> <th>Maximum + Actual</th> </tr> </thead> <tbody> <tr> <td></td> <td>0</td> <td>65280 28</td> <td>65535 0</td> </tr> <tr> <td>c</td> <td>1</td> <td>65280 28</td> <td>65535 0</td> </tr> <tr> <td>c</td> <td>2</td> <td>65280 0</td> <td>65535 0</td> </tr> <tr> <td>c</td> <td>3</td> <td>65280 0</td> <td>65535 0</td> </tr> </tbody> </table>		CSS ID	Devices in SS0	Devices in SS1		/ ID	Maximum + Actual	Maximum + Actual		0	65280 28	65535 0	c	1	65280 28	65535 0	c	2	65280 0	65535 0	c	3	65280 0	65535 0
	CSS ID	Devices in SS0	Devices in SS1																																														
	/ ID	Maximum + Actual	Maximum + Actual																																														
	0	65280 28	65535 0																																														
d	1	65280 28	65535 0																																														
d	2	65280 0	65535 0																																														
	3	65280 0	65535 0																																														
	CSS ID	Devices in SS0	Devices in SS1																																														
	/ ID	Maximum + Actual	Maximum + Actual																																														
	0	65280 28	65535 0																																														
c	1	65280 28	65535 0																																														
c	2	65280 0	65535 0																																														
c	3	65280 0	65535 0																																														

Even after *UserA* has successfully deleted the two channel subsystems, *UserB* sees the *Channel Subsystem List* with all channel subsystems, because this is the state of the IODF as loaded from storage. Let us suppose, he wants to change channel subsystems CSS 1, CSS 2 and CSS 3. An appropriate message is displayed for each channel subsystem that is already deleted. When returning to the *Channel Subsystem List*, *UserB* sees the refreshed contents of this panel.

<i>UserA:</i>					<i>UserB:</i>	
Channel Subsystem List					Message List	
Processor ID . . . : BBBX					Messages are sorted by severity. Select one or more, then press Enter.	
	CSS Devices in SS0		Devices in SS1			
/ ID	Maximum	+ Actual	Maximum	+ Actual	/ Sev	Msg. ID Message Text
_ 0	65280	28	65535	0	_ E	CBDA188I Channel Subsystem ID 1
- 3	65280	0	65535	0	#	does not exist for processor BBBX
					_ E	CBDA188I Channel Subsystem ID 2
					#	does not exist for processor BBBX

Viewing information

HCD offers several possibilities to view information that might be helpful when defining or maintaining your configuration data.

Viewing object definitions

To view information about objects that are defined in your configuration, select the *Define, modify, or view configuration data* option from the *Primary Task Selection* panel. From the resulting panel, select the desired object to get the appropriate list, on which you can use *Work with object* actions from the context menu that lead to further list panels that display other related objects. You can use these panels to define, modify, and view configuration data. See Figure 40 on page 76 on how to navigate to related list panels of different objects.

Viewing full definition of an object

You can ask HCD to display, for viewing only, the full definition of an object. This might include additional information that is not displayed on the action list panel.

To view object definitions select an object on an action list and use the *View object definition* action (or action code **V**).

This following lists offer this possibility:

- Processor List
- PCIe Function List
- Channel Path List
- Control Unit List
- I/O Device List

Viewing additional object lists

Besides the action list panels shown in Figure 40 on page 76, there are list panels, on which you can only view information about objects without being able to change it. You can navigate to these lists using the *View objects* actions from the context menu. HCD offers *View objects* actions on the following panels:

List Panel

View

Channel Path List

Connected switches

Partition List

Attached channel paths

Attached control units

Attached devices

Operating System List

Generics

Generics List

Devices

Esoteric List

Devices

Graphical view

HCD offers you the possibility to view a graphical representation of the configuration.

Use the task *Create or view graphical configuration report* on the *Primary Task Selection* panel to view the entire configuration. To view objects in context of their attached objects you can also select an object from an action list panel and use the *View graphically* action from the context menu (or action code **h**). The following object lists support this possibility:

- Channel Path List
- Control Unit List
- I/O Device List (only for devices that connect to a control unit)
- Partition List
- Switch List

For more information on how to view a graphical report, refer to “Create or view graphical configuration reports” on page 257.

Viewing logical control units

On a Control Unit List or I/O Device List, you can display the groups of *logical control units* that HCD has created to represent the physical control units defined in a configuration. Logical control units are used by the CSS to schedule the processing of I/O requests.

Viewing coupling facility information

HCD lets you view information for a specific CF channel path. You can view, for example, access and candidate list of selected channel paths and information on the CF control units and devices in a CF channel path connection.

On the CF Channel Path Connectivity List, select a channel path and one of the following actions:

- The *View source channel path definition* action displays the View Channel Path Definition panel for the source channel path of the CF connection.
- The *View destination channel path def.* action displays the View Channel Path Definition panel for the destination path of the CF connection.
- The *View CF control unit and devices* action displays the View CF Control Unit and Devices panel that shows the CF control unit number(s), the starting CF device number(s), and the range of devices defined for a CF connection.

Viewing CTC connections

HCD offers you the possibility to view and verify your CTC connections. You can view existing CTC connections including online diagnostic messages on the following lists:

- Processor List
- Partition List
- Channel Path List
- Control Unit List
- I/O Device List

The *CTC Connection List* lets you immediately verify whether your definitions are done correctly.

ESCON channel-to-channel support

An ESCON CTC connection requires a CTC channel at one end of the connection and a CNC or FCV channel at the other end of the connection. The two channels can be considered as communicating directly with each other in a peer-to-peer fashion. Each channel defines the channel at the other end of the CTC connection as an SCTC control unit. This is illustrated in Figure 69.

FICON channel-to-channel support

FCTC support differs from ESCON CTC support not only in the channel types used. The main differences are as follows:

- An FCTC connection is given via a FICON channel path on each side of the FCTC communication line.
- It is possible to have an FCTC connection between the LPARs of the same CEC via a single FICON channel path in a switched environment.

In order to be usable as an FCTC connection channel, a FICON channel path must be defined to an FCTC control unit which is connected to FCTC devices.

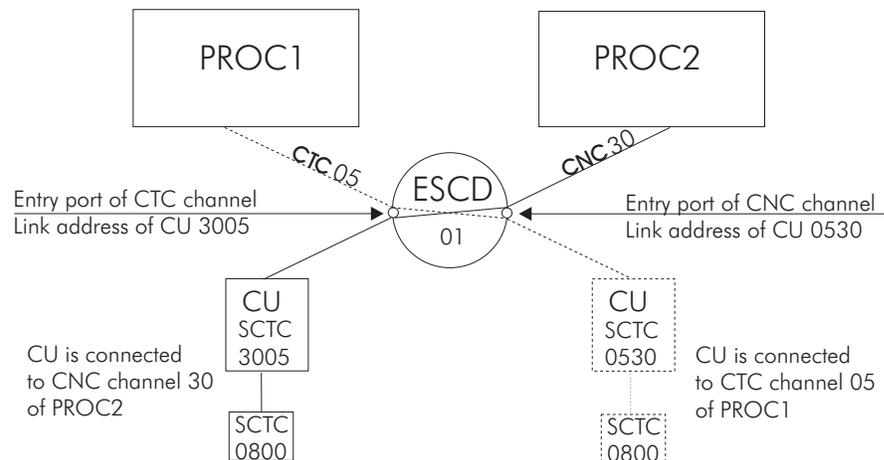


Figure 69. CTC/CNC connection established using a dynamic connection

The entry port of the channel at the other end of the CTC connection corresponds to the link address of the control unit representing the channel.

The CTC devices associated with the control units at both ends of the CTC connection may have different device numbers, but they must have the same unit address. The device type of both devices must be the same (for example, SCTC or BCTC).

Restrictions applying to the CTC Connection List

- HCD can only show CTC connections if the connected processors are defined in one IODF.
- For a switched SCTC connection, the CNC/FCV and CTC channel paths must be connected to the same ESCON director. CTC connections running via chained ESCON directors cannot be determined.
- CTC connections using a stand-alone CTC adapter cannot be shown.

CTC connections with shared channels

If your processor has MIF support you can share your channels among several partitions to save physical connections. The following figure shows you the case when a CNC channel is shared between two partitions. The CTC channel will need a separate control unit definition for each partition sharing the CNC channel. Each of these control unit definitions has the same destination link address but the control unit logical addresses (CUADD) must be different. The control unit logical address must correspond to the image number of the logical partition.

Note: If the target channel path is non-shared, either you must not specify a control unit logical address, or its value must be 0.

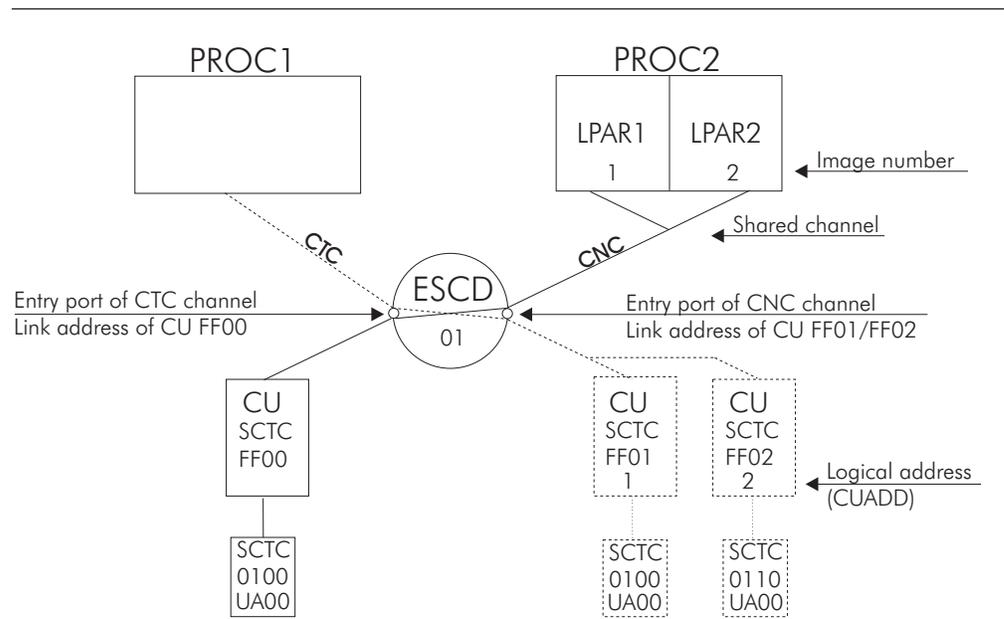


Figure 70. CTC/CNC connection established using shared channels

For further specification rules, refer to the *IOCP User's Guide* for your processor.

Point-to-point CTC connection

Each point-to-point CTC connection is realized by a cable plugged into the ends of the connection (imagine, the connecting cable has a serial number). In addition the control unit and device definitions must match the desired CTC connection – similar to the switched connection (CUADD on one side must match the partition image number on the other side, the explicit device candidate list must allow the LPAR of the same side to use the device).

To verify and report point-to-point CTC connections, HCD uses serial numbers. To find the both ends of the connection, all control units connected to the chpids must have the same serial numbers.

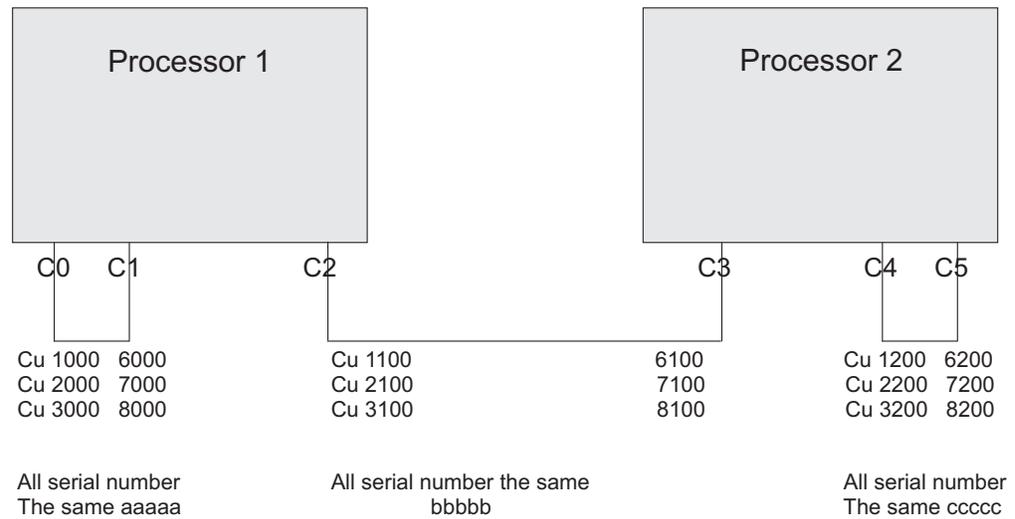


Figure 71. CTC connection established using point-to-point

How to view CTC connections

You can use action *View related CTC connections* (or action code **k**) on the following lists:

- Processor List
- Partition List
- Channel Path List
- Control Unit List
- I/O Device List

A panel similar to the following one is displayed:

```

Goto Filter Backup Query Help
-----
                CTC Connection List      Row 1 of 14 More:  >
Select CTC connections to view CTC Messages, then press Enter.

-----CTC or FC side-----  -----CNC/FCV or FC side-----
/ Proc.CSSID Part.  Devices CH CU  Proc.CSSID Part.  Devices CH CU  Msg.
- PROC001A          0500,5  20 1020 PROC001          0100,5  10 1010 G750
- PROC001A          0690,1  20 0069                                     G752
- PROC002           0650,1  11 0065 PROC002          0660,1  13 0066 G753
- PROC002           0680,1  11 0068                                     G752
- PROC002           0701,1  12 0050 PROC003   PART1   0301,1  10 1012
- PROC002           0800,5  22 0060 PROC003   PART2   0400,5  11 1013
- PROC002           0805,1  22 0060 PROC003   PART2   0405,1  11 1013 G751
- PROC002           0806,3  22 0060 PROC003   PART2   0406,3  11 1013
- PROC002           2400,1  24 0024                                     G756
- XMP1.1   PART1   1105,1  21 0105 PROC001          1106,1  10 0106
- XMP1.1   PART1   1107,1  21 0107 PROC002          1108,1  26 0108 G750
                                     PROC002          0200,2  10 1011 G757
                                     XMP1.1   PART1   0300,1  10 1012 G754
                                     XMP1.1   PART2   0300,2  10 1012 G754
  
```

This panel shows the definitions of the CNC/FCV side in relation to the definitions of the CTC side, such as processor, partition, channel path, control unit, and device information.

Incomplete CTC definitions: If the CTC connection is not correctly defined, the fields on the CTC Connection List can be incomplete and an error message is shown. For example, G754 in column Msg. refers to message CBDG754I, which indicates that HCD cannot determine the connection, because no control units and devices match to the processor, partition, control unit, and device of the same row.

Displaying more detailed information

Scroll to the right to see more detailed information about the CTC side of the connection, such as channel path mode, switch information, detailed control unit and device information.

```

Goto Filter Backup Query Help
-----
                CTC Connection List (CTC/FC)      Row 1 of 14 More: <>

Select CTC connections to view CTC Messages, then press Enter.

  -Partition--  ----Devices-----  -CHPID-  Entry Dyn      Link CU
/ Proc.CSSID Name    Num Number  Type OS  UA ID Mode SW PO SW  CU  Addr ADD
- PROC001A          0500,5   BCTC N  00 20 DED 05 F0 05 1020 E120
- PROC001A          0690,1   SCTC N  00 20 DED 05 F0 05 0069 E728
- PROC002           0650,1   SCTC N  00 11 DED 05 E7 05 0065 E840
- PROC002           0680,1   SCTC N  00 11 DED 05 E7 05 0068 F0
- PROC002           0701,1   SCTC N  01 12 DED 05 D0 05 0050 F4  1
- PROC002           0800,5   SCTC N  00 22 DED 05 D7 05 0060 F3  2
- PROC002           0805,1   BCTC N  05 22 DED 05 D7 05 0060 F3  2
- PROC002           0806,3   SCTC N  06 22 DED 05 D7 05 0060 F3  2
- PROC002           2400,1   SCTC N  00 24 DED 04 E0 05 0024 E0
- XMP1.1 PART1      1 1105,1   SCTC N  00 21 DED 05 F7 05 0105 E1
- XMP1.1 PART2      2 1107,1   SCTC N  00 21 DED 05 F7 05 0107 F8

```

Scroll once again to the right to see the same detailed information for the CNC/FCV side of the connection.

```

Goto Filter Backup Query Help
-----
                CTC Connection List (CNC/FCV/FC)  Row 1 of 14 More: <

Select CTC connections to view CTC Messages, then press Enter.

  -Partition--  ----Devices-----  -CHPID-  Entry Dyn      Link CU
/ Proc.CSSID Name    Num Number  Type OS  UA ID Mode SW PO SW  CU  Addr ADD
- PROC001          0100,5   BCTC N  00 10 DED 05 E1 05 1010 F0  0
- PROC002          0660,1   SCTC N  00 13 DED 05 E8 05 0066 E7
- XMP1.1 PART1      1 0301,1   SCTC N  01 10 SHR 05 F4 05 1012 D0
- XMP1.1 PART2      2 0400,5   SCTC N  00 11 SHR 05 F3 05 1013 D7
- XMP1.1 PART2      2 0405,1   SCTC N  05 11 SHR 05 F3 05 1013 D7
- XMP1.1 PART2      2 0406,3   SCTC N  06 11 SHR 05 F3 05 1013 D7
- PROC001          1106,1   SCTC N  00 10 DED 05 E1 05 0106 F7
- PROC002          1108,1   SCTC N  00 26 DED 05 F8 05 0108 F7  1
- PROC002          0200,2   SCTC N  00 10 DED 1011 C5
- XMP1.1 PART1      1 0300,1   SCTC N  00 10 SHR 05 F4 05 1012 D0
- XMP1.1 PART2      2 0300,2   SCTC N  00 10 SHR 05 F4 05 1012 D0

```

Filtering CTC definitions

To get a better overview of your CTC connections you can filter the list by specifying different filter criteria. Select action bar *Filter* and then *Set filter*. The following panel appears:

Filter CTC Connections

Specify or revise the following filter criteria and press Enter.

Message ID _____
Device type _____ (SCTC/BCTC/FCTC)
Dynamic switch _____ +

CTC or FC side	CNC/FCV or FC side
Processor.CSSID _____ +	Processor.CSSID _____ +
Partition _____ +	Partition _____ +
CHPID _____	CHPID _____

CU number _____	CU number _____
Starting device no. _____	Starting device no. _____
Defined to OS _____ (Y/N)	Defined to OS _____ (Y/N)

Printing CTC connection lists

You can also print the list panel using the SAVE command as described in “How to print list panels” on page 263.

Displaying diagnostic messages

For each connection, HCD displays one message, even if the connection includes several errors. HCD displays the messages according to the following priority list:

1. CBDG750I

Logical address (CUADD) is specified for CU @1, but CHPID @2 of processor @3 is not defined as shared.

2. CBDG751I

Device type of device @1 connected to processor @2, CHPID @3 does not match with device type of device @4 on the other side.

3. CBDG752I

Channel path type error. CHPID @1 of processor @2 is connected to a CHPID @3 of processor @4 with the same type.

4. CBDG753I

Wrap around connection detected for processor @1 (partition @2) via CHPID @3 and CHPID @4.

A message list may look as follows. The messages are sorted by severity.

Save Query Help

Message List Row 1 of 11
Command ==> _____ Scroll ==> PAGE

Messages are sorted by severity. Select one or more, then press Enter.

```
/ Sev Msg. ID Message Text
_ E   CBDG750I Logical address (CUADD) is specified for CU 1010,
#     but CHPID 20 of processor PROC001A is not defined as
#     shared.
_ E   CBDG752I Channel path type error. CHPID 20 of processor PROC001A
#     is connected to CHPID 11 of processor PROC002 with the
#     same type.
_ W   CBDG753I Wrap around connection detected for processor PROC002
#     (partition - none -) via CHPID 11 and CHPID 13.
_ I   CBDG756I HCD cannot determine connection. CHPID 24 of processor
#     PROC002 is connected via chained switches.
```

Chapter 7. How to work with switches

Overview

This information unit explains:

- The possibility of switch connections
- The advantages when you define switches with HCD
- How to work with switches (defining, changing, priming, deleting)
- How to work with connections to switches (channel paths, control units, priming switch port names and connections, actions on the Port List)
- How to work with switch configuration data (defining, changing, deleting)
- How to migrate, activate, and save switch configuration data

You can define switches, switch connections and how the switches are physically cabled. A switch configuration, also called port matrix, defines how the various ports of the switch connect to each other. For example, the switch configuration defines whether a port is blocked, has a dedicated connection to another port, or whether dynamic connections to other ports are allowed or prohibited. In other words, the switch configuration defines the inside of a switch.

Possibilities of switch connections

ESCON switches (aka ESCON Directors, ESCDs) enable either dynamic connections or dedicated connections. FICON switches only allow the definition of dynamic connections.

HCD supports fabrics containing cascade switching using FICON switches. Such a fabric consists of two or more FICON switches.

Note: Other than for chained ESCON switches where the dynamic switch for a channel path specifies the switch containing the port address that is used as link address, in the FICON case the dynamic switch ID is always the channel path entry switch.

The following four figures illustrate the path types of switch connections for ESCON Directors or FICON switches. For FICON switches, only the first and the fourth configuration type is supported. Figure 72 on page 164 shows a configuration with a single switch; the entry switch is the dynamic switch. (The dynamic switch in HCD corresponds to the SWITCH keyword of the CHPID macroinstruction from the IOCP point of view.)

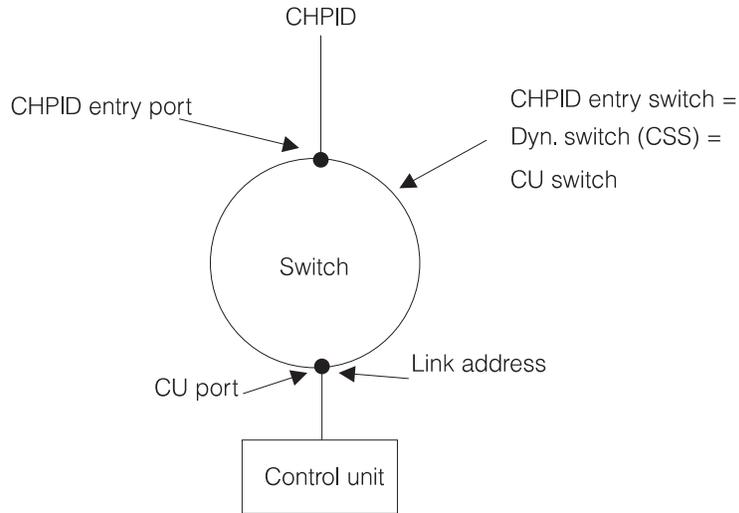


Figure 72. Configuration with one switch

Figure 73 shows a configuration with two switches, where the entry switch is different from the dynamic switch. (The two switches are chained and the entry switch for the channel path has a dedicated connection.)

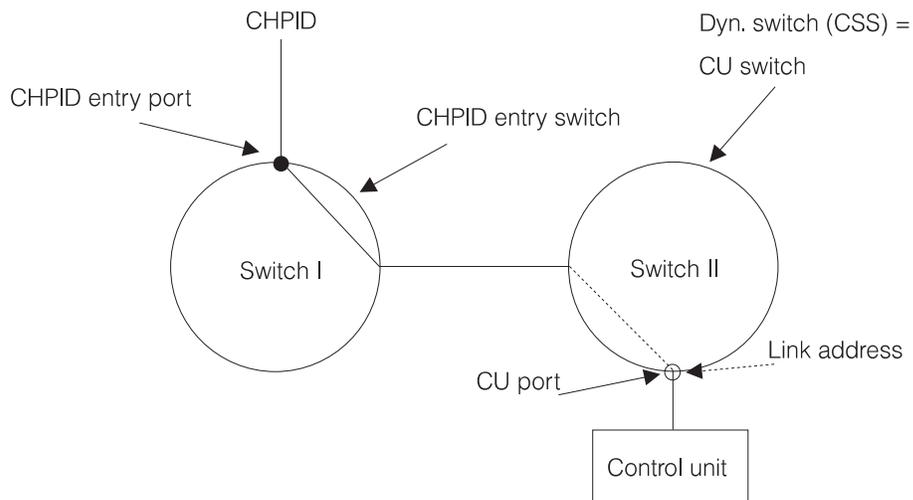


Figure 73. Configuration with two switches (1)

Figure 74 on page 165 shows a configuration with two switches, where the entry switch is the same as the dynamic switch. (The two switches are chained and the CU switch has a dedicated connection.)

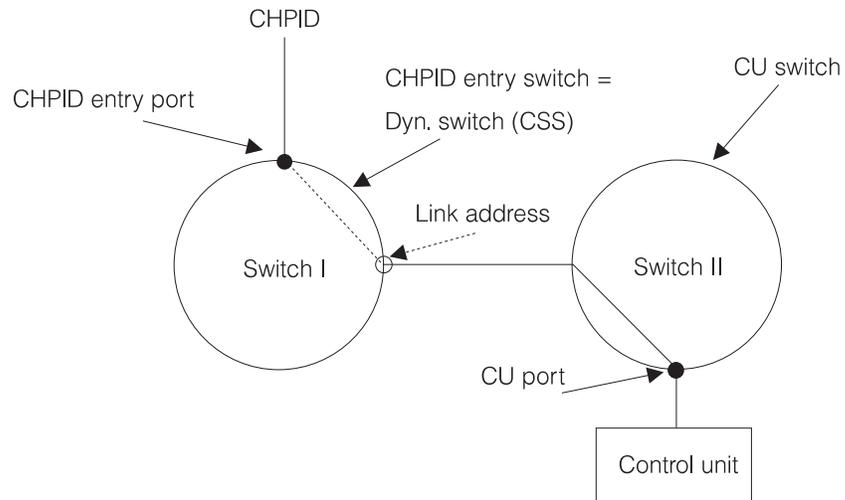


Figure 74. Configuration with two switches (2)

Figure 75 shows a configuration with cascading switches. The fabric in this figure contains two cascading FICON switches. The link address 5904 specifies 59 as switch address and 04 as port address.

Note:

The switch address is unique within a fabric, but may occur also in other cascaded switch fabrics. However, as HCD has no knowledge of which switches are within the same fabric, it is highly recommended to assign unique switch addresses across fabrics, for example, by using the switch IDs as switch addresses.

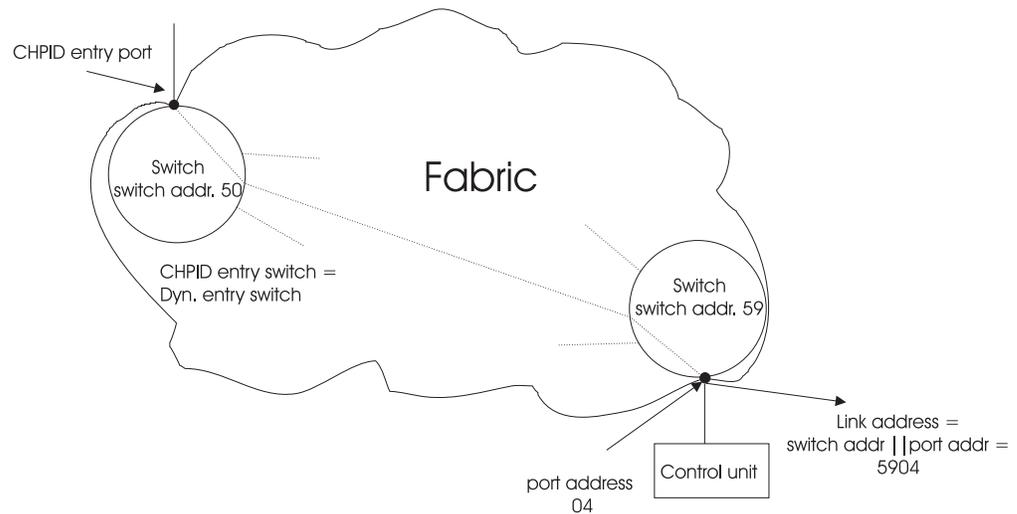


Figure 75. Configuration with two cascading FICON switches

Advantages of switch definitions with HCD

You have the following advantages when you define switches with HCD:

- More rigorous validation of the configuration definition.

If all switches between the channels and control units are defined, HCD can determine whether a valid path exists between the processor and control unit. For example, HCD can validate that the destination link address specified for a channel path is a valid port on the dynamic switch.

- The possibility to define several switch configurations for each switch.
- The possibility to define and activate switch configurations without leaving HCD (from the same *workplace*).

If you have defined a switch configuration using HCD, you can activate the switch configuration without leaving HCD and do not need any knowledge about other operating system components that are involved in the activation process.

- The possibility to define a switch as CU, device, and switch as such, to:
 1. Migrate an active switch configuration from a switch, or a saved switch configuration from a switch file, or convert an ISPF table to a HCD switch configuration for later manipulation by HCD.
 2. Send switch configuration data from an IODF to a switch, making it the active port matrix, or save it in a switch file.
- Graphical configuration reports include switch connections.

Note: HCD supports a generic FICON switch (type FCS) supporting port addresses 00 to FF. This switch type does not support a switch control unit and switch device and therefore can not be accessed by I/O Operations functions like migrate or activate switch configuration.

Defining switches

To define switches and their associated ports, you need to

- define switch characteristics,
- define connections to channel paths, control units, and other switches,
- define switch configuration data (port matrix).

Working with switches

In this section, you can learn how to define, change, prime, and delete switches.

Defining switch characteristics

1. On the *Primary Task Selection* panel, select *Define, modify, or view configuration data* and on the resulting panel the object *Switches*. HCD displays the list of all switches currently defined in the IODF.

```

Goto Filter Backup Query Help
-----
Switch List                               Row 1 of 6 More:  >
Command ==> _____ Scroll ==> PAGE

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

/ ID Type +      Ad Serial-# + Description          CU  Dev
_ 01 2032        21 10145      Switch 01          0701 0701 >
_ AA 9032        _  10146      Switch AA          001A 001A >
_ AB 9032        _  10147      Switch AB          001B 001B >
_ AC 9032        _  10678      Switch AC          001C 001C
_ AD 9032        _  10679      Switch AD          001D 001D
_ AE 9032-3      _  20995      Switch AE          001E 001E
***** Bottom of data *****

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

```

Figure 76. Switch List (left part)

The Switch List (left part), Figure 76, lists one switch control unit and device. If there is more than one switch control unit and device, the list entry gets an indication ('>'). With the F20=Right key, you can scroll to the right part of the Switch List. Up to five switch control units and devices can be shown. If there are more, an indication is given for the corresponding entry ('Yes' in column 'More?' on the right part of the Switch List). These additional switch control units and devices can be viewed, for example, on the Port List for port FE.

```

Goto Filter Backup Query Help
-----
Switch List                               Row 1 of 6
Command ==> _____ Scroll ==> PAGE <

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

/ ID Type +      CU  Dev  CU  Dev  CU  Dev  CU  Dev  CU  Dev  More?
_ 01 2032        0701 0701 0702 0702 0703 0703 0704 0704 0705 0705 Yes
_ AA 9032        001A 001A 002A 002A 003A 003A
_ AB 9032        001B 001B 002B 002B 003B 003B
_ AC 9032        001C 001C
_ AD 9032        001D 001D
_ AE 9032-3      001E 001E
***** Bottom of data *****

F1=Help    F2=Split    F3=Exit    F4=Prompt    F5=Reset    F7=Backward
F8=Forward  F9=Swap    F10=Actions F11=Add      F12=Cancel  F13=Instruct
F19=Left   F22=Command

```

Figure 77. Switch List (right part)

2. Use the F11=Add key to add a new switch.

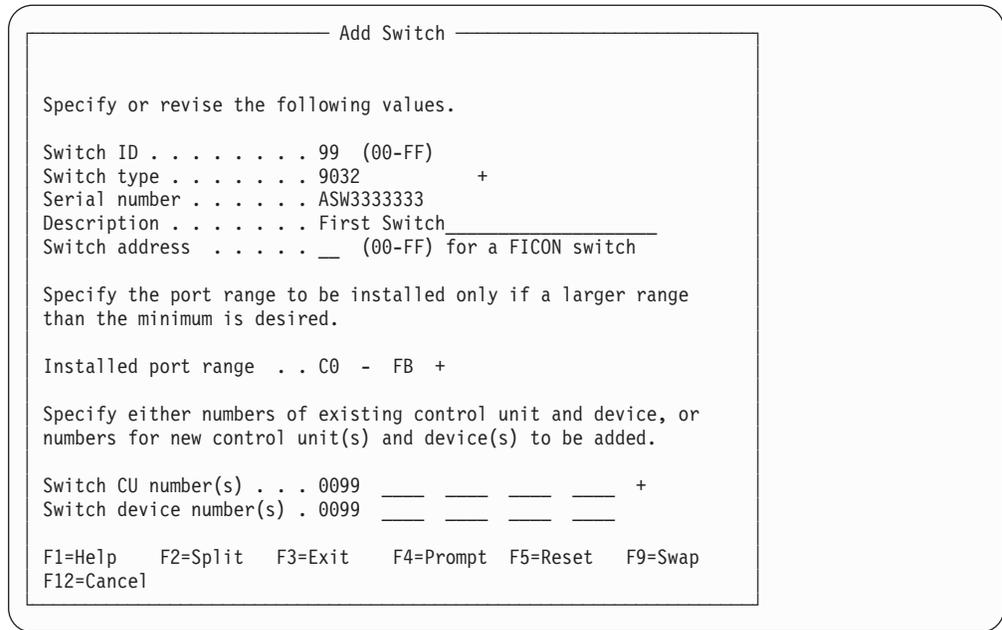


Figure 78. Add Switch

HCD allows you to specify the port range of a switch to be set to installed, if more ports are to be used than the minimum range. Specify the first and last port of the range you want to use. If you do not specify values for the Installed port range field, the hardware status of the minimum range of supported ports is set to installed.

In order to allow consistency checks for the configuration, when adding a new switch, you can optionally define a switch address for a FICON switch.

You can also specify control unit numbers and device numbers for the switch.

On the Add Switch panel you can initially define up to five switch control units and devices for the switch. To define more than five switch control units and devices, or to add additional switch control units and devices later, you must use the control unit and device definition dialogs.

If you specify switch control units that do not yet exist, they are automatically added as new objects to the IODF, and are connected to the switch through the switch control unit port. In this case, you need to specify new switch devices. The switch devices are also automatically added as new objects to the IODF and connected to the switch through the switch control units. However, to complete the configuration path, you must attach the switch control units and switch devices to a processor, and then you can assign the switch devices to an operating system.

If the switch control units already exist, they are automatically connected to the control unit port on the newly defined switch. In this case, you do not need to specify switch devices. If you do, the switch devices must already exist and be attached to the designated switch control units.

Specified serial numbers or descriptions are also copied to the switch control units and switch device definitions.

3. After you press the Enter key, HCD displays the updated Switch List.
 - Connect the switch control units to the processor (which also connects the switch devices to the processor). To specify additional parameters use the **Change** action on the Control Unit List. For details, see “Defining processor attachment data” on page 128.

- Connect the switch devices to the operating system. Use the *Change* action on the I/O Device List. For details, see “Changing devices” on page 141.

Changing switch data

To change the following switch characteristics, you can type over the columns on the Switch List, or you can use the *Change* action from the context menu (or action code **c**) on the Switch List:

- Switch type
- Serial number and description

When you change the type, serial number, or description of the switch, the control units, and devices attached to the switch are also updated.

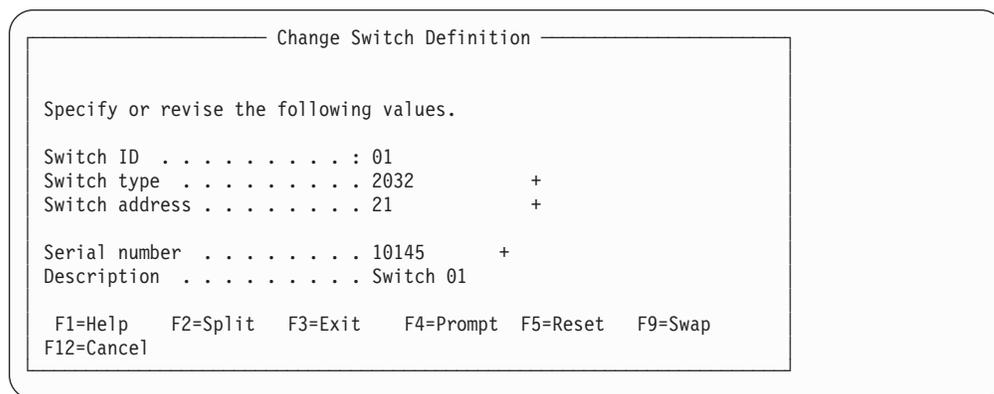


Figure 79. Change Switch Definition

Priming switch data

You can prime your I/O configuration in a work IODF with the switch serial number for the active processor. For the prerequisites for this function see “Prerequisites” on page 9

To prime, select the action *Prime serial number* from the context menu (or action code **i**) on the Switch List.

The Confirm Priming Switch List shows the selected switches with the sensed data for the switch types, serial numbers, switch control units, and switch device numbers, and their corresponding definitions in the IODF.

Note: Only one defined switch control unit and one defined switch device is shown even if several have been defined. If the sensed switch control unit and device is one that has been defined, it will be displayed. Otherwise the defined switch control unit and device with the lowest control unit number will be displayed.

An update of the switch serial number also updates the serial number of the corresponding switch control units and switch devices.

The sensed data for the switch serial numbers are shown on the Confirmation panel, and can be accepted, or rejected before being incorporated into the IODF. If a value is blanked out, the defined IODF value is not changed. If you use the F12=Cancel key, none of the sensed values is used.

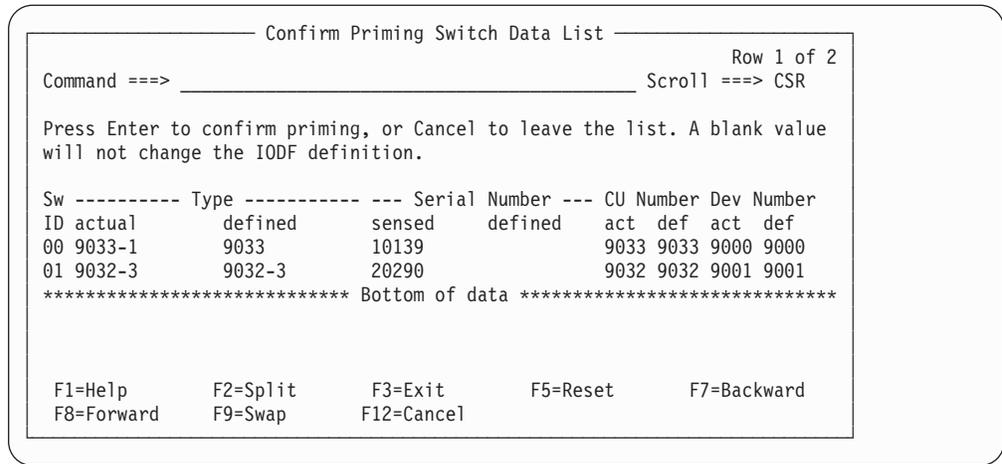


Figure 80. Confirm Priming Switch Data List

Moving ports

When selecting *Move Ports*, HCD offers the possibility of moving control unit, channel path or switch port connections on the same switch or from other switches to the selected target switch.

To perform this action, select the target switch for a port move action on the Switch List (see Figure 76 on page 167). The panel Actions on selected switches appears and the action *Move Ports* can be selected. You can also reach this panel directly by typing in the action code **x** next to the appropriate switch in the Switch List menu. This brings up the following panel Move Ports to a Target Switch.

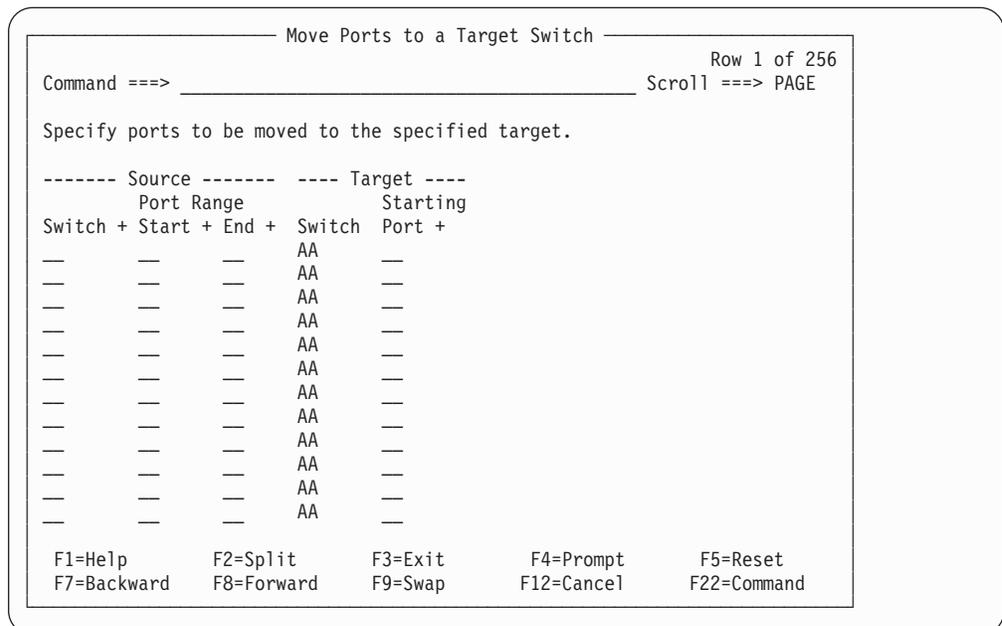


Figure 81. Move Ports to a Target Switch

This panel contains data entry fields for the ports to be moved. It is also possible to move a range of ports from a switch to the target switch occupying subsequent

port addresses starting with the target port address specified. The target switch field in this panel has been preset and cannot be changed.

Depending on the context, HCD performs the following as part of the *Move Ports* action:

- Copies the attributes of the source port to the target port (and set the target port to installed, if necessary).
- Disconnects all source ports from the connected units.
- Connects all target ports to the units previously connected to the source ports.
- Copies existing port configurations of the source port if the move is on the same switch and port configurations exist. The source port configurations will be set to default, i.e., all dynamic and dedicated connections are reset.
- Changes the dynamic switch of the connected channel path to the target switch if the source switch serves as a dynamic switch and the target switch is different from the source switch.
- Changes the link address to the target port if the source port serves as a link address to a channel path connection for a control unit and is connected to a control unit or another switch.

Note: HCD does not perform any checks on whether the user also moves implied ports as well. For example, if a channel path is moved to another switch, the control units that are connected to the channel path must also be moved. Moving a control unit may imply that connected channel paths must be moved as well. If not all implied ports are moved, the configuration may become invalid and a validation error will be shown either during the *Move Ports* action or later during the *Build Production IODF* action.

Note: If the target switch has switch configurations defined and the port move occurs between different switches, then the switch configurations must be adapted after the port move action.

The *Move Ports* action is especially of value when consolidating switches or installing new switches.

Deleting switches

You can delete the complete definition of a switch or switch configuration by using the *Delete* action from the context menu (or action code **d**) on the Switch List or Switch Configuration List. This also deletes the connections from the ports to channel paths, control units, and other switches. The link address and dynamic switch definitions for a channel path are not deleted.

The Confirm Delete Switch panel shows all the switch control units and devices that will be deleted with the switch. If you don't want them to be deleted with the switch, disconnect the control units from port FE of the switch, before you delete the switch.

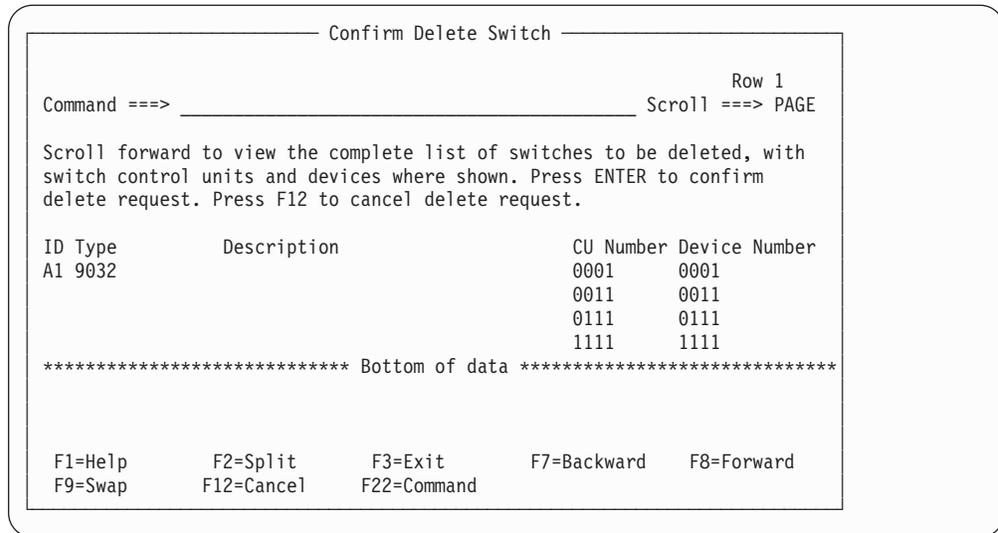


Figure 82. Confirm Delete Switch

Working with connections to switches

In this section, you will learn about how to define connections to switches, for example channel paths, control units, other switches, how to prime switch port data, and what kind of additional actions is offered on the Port List.

Defining connections to switches

You can connect the following objects to a switch starting from the Switch List:

- Channel paths
- Control Units
- Other switches

You can also define connections from the objects to the switch when defining the object themselves. See Chapter 6, “How to define, modify, or view a configuration,” on page 73 for a description how to define the objects.

Connecting a channel path

The following procedure describes how to define a connection between a channel path and a switch starting from the Switch List.

1. On the Switch List, select the switch and the *Work with ports* action from the context menu (or action code **p**). The Port List is displayed.

Ports which show value Y in column 0 indicate that they are occupied by a processor, control unit or switch that is not defined in the accessed IODF.

```

Goto Filter Backup Query Help
-----
Port List                               Row 1 of 61
Command ==> _____ Scroll ==> PAGE

Select one or more ports, then press Enter.

Switch ID . . . . . : 99  Address :      Switch AB
Switch configuration ID : SWAB

-----Connection-----
/ Port H Name +                Unit ID      Unit Type      O
- C0  Y _____
- C1  Y _____
- C2  Y _____
- C3  Y _____ PR CTC01      CHP 34 9672-R21  N
- C4  Y _____
- C5  Y _____
- C6  Y _____ PR MCSSPR01.0 CHP 18 2084-C24
# C6  PR MCSSPR01.1 CHP 18 2084-C24
- C7  Y _____
- C8  Y _____
- C9  Y _____
- CA  Y _____
F1=Help   F2=Split   F3=Exit   F4=Prompt   F5=Reset   F7=Backward
F8=Forward F9=Swap    F10=Actions F12=Cancel  F13=Instruct F22=Command

```

Figure 83. Port List

2. Select a port and the *Connect to channel path* action from the context menu (or action code **p**).

Occupied ports cannot be connected. However, you may change the Occupied status of a port by overwriting the Occupied indicator.

On the resulting Connect to Channel Path panel specify the target processor ID and channel path ID.

```

----- Connect to Channel Path -----

Specify the following values.

Switch ID : 99 Port . . . : C4

Processor ID . . . . . _____ +
Channel Subsystem ID . . _ _____ +
Channel path ID . . . . _ _____ +

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

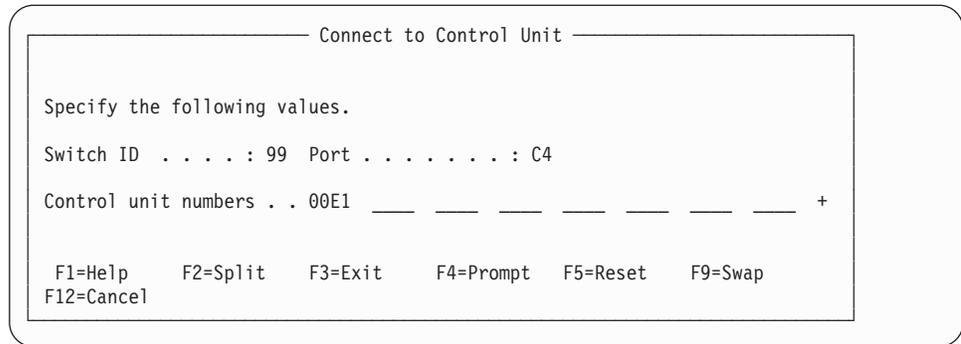
```

Note: If you selected a spanned physical channel path as connection target, HCD connects the port to all of the channel's instances across all channel subsystems. You can see the result of your connection action in the Port List (Figure 83).

Connecting a control unit

The following procedure describes how to define a connection between a control unit and a switch starting from the Switch List.

1. On the Switch List, select the switch and the *Work with ports* from the context menu (or action code **p**). The Port List is displayed (see Figure 83).
2. Select a port and the *Connect to control unit* action from the context menu (or action code **u**).



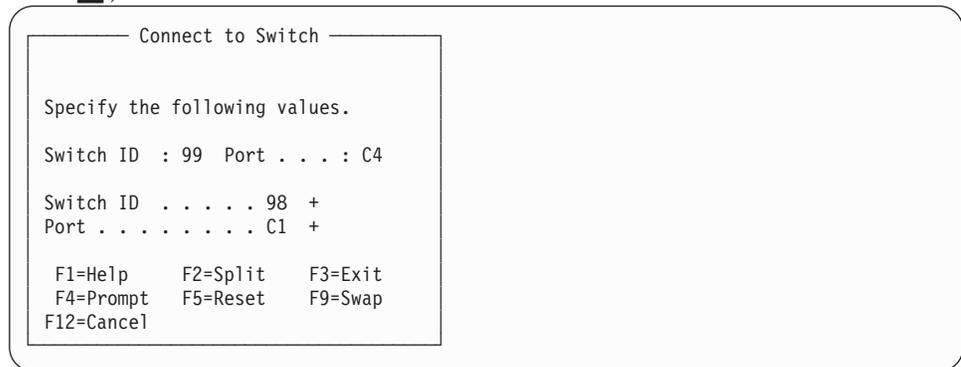
You can enter up to eight control unit numbers each time the panel is displayed.

3. Repeat defining connections for all control units connected to the switch.

Connecting another switch

The following procedure describes how to define a connection between a switch and another switch.

1. On the Switch List, select the switch and the *Work with ports* action from the context menu (or action code **p**). The Port List is displayed (see Figure 83 on page 173).
2. Select a port and the *Connect to switch* action from the context menu (or action code **w**).



3. Repeat defining connections for all other switches connected to the selected switch.

Priming switch port data

You can prime your I/O configuration in a work IODF with the switch port names, and connections for the active processor. For the prerequisites for this function refer to "Prerequisites" on page 9.

To prime, select the action *Prime port name and connections* from the context menu (or action code **i**) on the Port List.

The Confirm Priming Port Data List lists the selected ports with the sensed data for the port names, and for the connected control units, or switches, or the connections to channel paths of the active processor. Their corresponding definitions in the IODF are shown in the line under the sensed data on the panel.

You get sensed data for connected channel paths only if the processor definition in the IODF contains a serial number that matches the serial number of the active processor.

The sensed port names and connection data can be confirmed before being taken into the IODF. If a value is blanked out, the defined IODF value is not changed. If you use the F12=Cancel, or the F3=Exit key, none of the sensed values is used.

```

Confirm Priming Port Data List
Row 1 of 4
Command ==> _____ Scroll ==> CSR

Press Enter to confirm priming, or Cancel to leave the list. A blank value
will not change the IODF definition.

Switch ID . . : 01
Port --- Sensed Port Name --- ----- Sensed Connection -----
  -- Defined Port Name --- ----- Defined Connection -----
A1  200A-E                      CU  200A          3990-6          1
   200A-E
A2  400B-CG                      CU  520B          3990-3          2
   400B-CG
A3  360A-00                      CU  360A          9343-DC4        3
   VMA(32)
A4  VMA(32)                      CU  360A          9343-1          4
   VMA(32)
   PR  VMABASIC CHPID 32 9672-R61
***** Bottom of data *****
F1=Help      F2=Split    F3=Exit     F5=Reset    F7=Backward
F8=Forward   F9=Swap     F12=Cancel

```

- 1** No defined data is available for the port connection on the active system. The sensed and defined port names are the same.
- 2** The sensed and defined port names are available. The port connection is defined in the IODF, but no sensed data is available on the active system.
- 3** No sensed port name data is available on the active system. The port connection shows differences of the switch type, but the sensed data is not taken in the IODF.
- 4** No sensed data is available for the port connection on the active system. The sensed and defined port names are the same.

Additional actions on the port list

Besides connecting channel paths, control units, and other switches to a switch, you can perform additional actions on the Port List. Most of these actions are also possible on the Port Matrix panel, this panel offers an alternative procedure. The following panel shows data after connecting a channel path, a control unit, and another switch as described in the previous sections.

```

Goto Options Filter Backup Query Help
-----
Port List                               Row 22 of 128
Command ==> _____ Scroll ==> CSR

Select one or more ports, then press Enter.

Switch ID . . . . . : 71   Address :      Switch 71
Switch configuration ID : SW71

/ Port H Name +          -----Connection----- DED
      Y          Unit ID      Unit Type      O B CON +
- 92 Y          SW 03        PO 20 2032        N N -
- 93 Y          PR P2084.1   CHP 21 2084-B16      N N -
- 94 Y          PR P2084.2   CHP 22 2084-B16      N N -
- 95 Y SUBC6F-A2          N N -
- 96 Y SUBC91-A0          N N -
- 97 Y          N N -
- 98 Y          N N -
- 99 Y SUBC6F-C0          CU BA80          3990      N N -
# 99          CU BD80          3990
- 9A Y SUBC91-A2          N N -
- 9B Y N40-50            N N -
- 9C Y SUBC2D            N N -
- 9D Y SUBC6F-C2          N N -
- 9E Y SUBC6F-H0          N N -
- 9F Y SUBC6F-H2          N N -
- A0 Y JE0-25            PR CB89          CHP 25 9672-E08      N N -
- A1 Y JE0-26            PR CB89          CHP 26 9672-E08      N N -

```

A disabled marker # in the action entry field indicates that the field is nonselectable and the whole row is disabled for processing. This occurs if more than one object is attached to one port, for example, for spanned channels or when multiple control units are connected to the same port. Except for the first object attached to the port, all other objects are flagged with the # sign. For example, if the port is connected to more than one unit, a disconnect action specified in the selectable row of that port will lead to the display of another panel where you can select the unit(s) to be disconnected.

Changing ports to installed or uninstalled

The H column indicates whether the ports are installed (Y for Yes) or not (N for No). If you did not specify a range of ports to be set to installed while adding a switch, HCD automatically sets the minimum range of ports to installed at switch definition time.

You can set the port to installed or uninstalled by just typing over the Y or N value in the H column.

Changing ports to occupied or not occupied

The O column indicates whether a port is occupied (Y for Yes) or not (N for No) by a system external to the IODF.

You can change the port to Occupied or Not Occupied by just typing over the Y or N value in the O column.

Establishing dedicated connections and blocking ports

If no switch configurations are defined for a switch, the B and Ded Con columns for blocked indicator and port of dedicated connection are not shown. If switch configurations are defined, data of the first switch configuration (in alphabetical order) is displayed. To display this data for other switch configurations, use *Select other switch configurations* from the *Option* action bar choice.

You can change the blocked indicator and dedicated connections for the switch configuration displayed in the panel header by just typing over the values in the appropriate column. See also “Establishing dedicated connections” on page 178 and “Blocking ports” on page 178 for detailed explanations.

Note: You cannot establish dedicated connections for a FICON switch.

Working with switch configurations

In this section you can learn about defining switch configuration data, changing the switch configuration IDs, and deleting switch configurations.

Defining switch configuration data

After defining the switch, you can define the switch configuration, that is the "inside" of the switch.

1. On the Switch List, select the switch and the *Work with switch configurations* action from the context menu (or action code **S**). HCD displays the Switch Configuration List containing all currently defined configurations for that particular switch.

```

Goto Backup Query Help
-----
                          Switch Configuration List                      Row 1 of 1

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

Switch ID . . . . . : 99          First switch

  Switch      Default
 / Config. ID Connection + Description
_ NIGHT      Allow
  
```

2. Use F11=Add to add a new switch configuration. The data-entry fields are shown, with sample data:

```

----- Add or Repeat Switch Configuration -----

Specify or revise the following values.

Switch ID . . . . . : 99
Switch configuration ID . BASIC__
Description . . . . . _____
Default connection . . . _1 1. Allow
                          2. Prohibit
  
```

The Default connection field sets the default connection for all ports, either allowed or prohibited. Individual port connections can be reset on the Port Matrix panel described in the next step.

3. On the Switch Configuration List, select the switch configuration and the *Work with port matrix* action from the context menu (or action code **S**). HCD displays the Port Matrix panel showing all ports currently installed on the switch.

```

Goto Backup Query Help
-----
Port Matrix                               Row 1 of 28 More: >
Command ==> _____ Scroll ==> PAGE

Select one or more ports, then press Enter.

Switch ID . . . . . : 99
Switch configuration ID . : BASIC      Default connection : Allow

/ Port Name +          Ded   --Dynamic Connection Ports Ex--
- E0 _____      B   Con + 0 1 2 3 4 5 6 7 8 9 A B C D E F
- E1 _____      N   -   * \ * * * * * * * * * * * * * * *
- E2 _____      N   -   ** \ * * * * * * * * * * * * * *
- E3 _____      N   -   * * * \ * * * * * * * * * * * *
- E4 _____      N   -   * * * * \ * * * * * * * * * * *

```

Scroll to the right on the Port Matrix panel to view further ports.

Establishing dedicated connections

You can establish a dedicated connection between two ports by specifying the number of a port to which a dedicated connection is defined in the Ded Con column. After pressing the Enter key HCD completes the definition by mirroring the definition. For example, if you specify a dedicated connection in the row of port E0 to port E4, HCD establishes the same dedicated connection in the row of E4 to port E0.

A dedicated connection acts like a physically cabled connection between two ports. Establishing a dedicated connection is not the same as prohibiting all but one connection to a port. Dedicated connections are required to support communication through an ESCD with an ESCON Converter (ESCC), and to support chained ESCDs.

If you maintain switch configurations with HCD, you must define the required dedicated connections before you connect a CVC or CBY channel path to a switch port.

If you have alternate required dedicated actions in a configuration (for example, for backup purposes), you must define alternate switch configurations.

Blocking ports

You can block or unblock a port by just specifying a Y for Yes or N for No in the B column.

Dynamic connection ports

In our example the default switch connection is set to allowed. To prohibit a switch connection from, for example, EA to FA, scroll down to port EA and scroll right to port FA. Then type over the * symbol with a p under the heading Dynamic Connection Ports. After pressing the Enter key HCD automatically mirrors the entries on the diagonal of the matrix. That means, HCD applies the same entry not only to the matrix element EA/FA but also to FA/EA.

To ensure a correct mirroring of the entries, press the Enter key each time you changed one complete row.

The following symbols can be shown under the heading Dynamic Connection Ports:

This is useful, if channel path - control unit connections have been added or changed and you want to define a switch configuration which considers all defined logical paths running through the selected switch.

To generate a matrix, perform the following steps:

1. On the Switch List select the switch and select the *Work with switch configuration* action from the context menu (or action code **s**). HCD displays the Switch Configuration List.
2. On the Switch Configuration List select a configuration (or action code **g**). HCD will issue an informational message requiring you to confirm your action. The successful generation of the matrix is confirmed by HCD.

When generating the switch configuration, all defined logical paths will be analyzed by HCD. In the case of chained connections, all possible paths of chaining switches will be determined by their respective switch configurations. Therefore, it is necessary, that you select a switch configuration for each chained switch, for which more than one switch matrix has been defined. In this case, the following panel

```

                                Select Active Switch Configurations                                Row 1 of 2
Command ==> _____ Scroll ==> PAGE

For each switch select one to be used as context to generate a switch
matrix.

Switch ID: 01 Configuration ID: SC1

/ Switch Configuration Description
_ 02      SC1
_ 02      SC2
***** Bottom of Switch Configurations data *****

```

will be displayed allowing selection of the related switch configurations which are considered for the *Generate matrix* action.

The following rules are applied when generating a matrix:

- The default connection for the switch configuration will be set to prohibited.
- A logical definition between a channel path using the switch as a dynamic switch and a control unit will lead to a dynamically allowed connection between the channel path entry port and the port serving as the link address.
- A logical definition between a channel path without a dynamic switch and a control unit, or connections via a chaining switch, will lead to a dedicated connection.

HCD will define a dedicated connection in all the cases where there is only one connection possibility left between the control unit and the channel path entry port. For dedicated connections, HCD considers only those ports, which have not yet been used for dynamic connections. In the case of chained connections, the selected switch configurations of chaining switches are used to determine the possible paths.

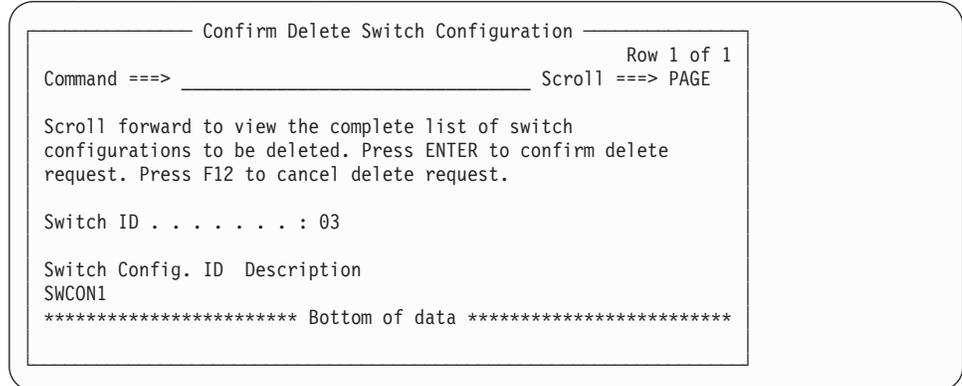
- Connections between ports connected to a channel path type, which make a specific port configuration necessary, will be set accordingly. For example, connections between ports serving as entry ports for FCV channels will be defined as explicitly dynamically prohibited.
- In addition to generating the matrix, HCD issues informational messages for all cases in which a path was not completely defined or where different paths

conflict with each other. If HCD is not able to determine port connections unambiguously, it will leave those definitions to the user.

Deleting switch configurations

To delete a switch configuration, perform the following steps:

1. On the Switch List, select the switch and select the *Work with switch configuration* action from the context menu (or action code **s**). HCD displays the Switch Configuration List.
2. On the Switch Configuration List select the switch configuration and the *Delete* action from the context menu (or action code **d**). The Confirm Delete Switch Configuration panel is displayed.



3. Press the Enter key to confirm deletion of the switch configuration, or use the F12=Cancel key to cancel the delete request.
4. The updated Switch Configuration List is displayed.

Migrating existing switch configurations

HCD allows to migrate a switch configuration from three sources into the IODF:

- An ISPF table containing a switch configuration as stored by I/O Operations
- An active switch
- A saved switch file

Prerequisites

To migrate from an ISPF table, the I/O Operations ISPF table data set name `escm.SINGITBL` has to be concatenated in the table library chain in your HCD start-up procedure, where “escm” is the high-level qualifier for your I/O Operations installation, or any other ISPF table data set containing ISPF tables saved by I/O Operations.

For additional information, see “Prerequisites” on page 9.

Migration steps

1. On the *Primary Task Selection* panel select *Migrate configuration data*.
2. On the resulting panel select *Migrate switch configuration data*.

```

Migrate Switch Configuration Data

Migrate switch configuration definitions to:

Switch ID . . . . . 98      +
Switch configuration ID . . . . . A98BC__ +

From one of the following:

__ 1. ISPF table
   2. Active Director
   3. Saved Director file

```

Specify the switch ID and the switch configuration ID of the empty switch configuration in the IODF to which you want to migrate the data.

If you have not previously defined the switch and the switch configuration, a panel appears that lets you define them (see “Defining switches” on page 166).

Select the source from which you want to migrate the switch configuration.

3. A panel appears on which you can define the source:
 - From ISPF tables:
 - Specify the name of the ISPF table that contains the configuration.
 - From an active director:
 - Specify the device number of the switch.
 - From a saved director file:
 - Specify the name of the switch file.
 - Specify the device number of the switch from which the switch file is to be taken.

The following panel is displayed when you are migrating from a saved switch file.

```

Migrate from Saved Director File

Specify the following values.

Director file name . . . . _____
Director device number . . ____

```

Note: The source of the switch configuration as specified in the switch device number field does not have to be the switch as specified in the Switch ID field on the Migrate Switch Configuration Data panel. It is possible to take a switch configuration from any switch and save it with HCD for another switch. However, the description of each port connection is saved with the switch configuration, and has to be updated to reflect the real channel path and control unit connections of the target switch.

Changing port names

In HCD, port names are saved with the ports, because the port names reflect the connections of a port. When migrating switch configurations with different port names to HCD, the IODF reflects the port names defined in the switch configuration that were migrated last.

Changing hardware status of a port

If the migration source contains ports set to installed and the existing IODF contains the same ports set to uninstalled, the hardware status after the migration depends on the kind of source:

Migration Source	Hardware Status of Ports after Migration
Active switch	Changed to installed
Saved director file	Uninstalled, migrated values ignored
ISPF table	Uninstalled, migrated values ignored

Activating switch configuration data

You can activate a switch configuration, which has been defined or changed with HCD, for a switch. Thus, change and maintenance in a configuration controlled by I/O Operations is simplified by the possibility to use the same user interface.

Prerequisites

- The IODF has to be a production IODF.
- A switch control unit must be defined for each switch.
- A switch device must be defined for each switch control unit.
- The switch control unit must have at least one channel path connected using the switch.
- For activation and saving, an ESCON Manager lock of another user must not exist.

For additional information, see “Prerequisites” on page 9.

Activation steps

1. On the *Primary Task Selection* panel, select *Activate or process configuration data* and from the resulting panel, select *Activate switch configuration*. The Activate Switch Configuration panel is displayed.

— Activate Switch Configuration —

Specify the following values, and select how to handle an existing ESCON Manager lock.

IODF name : 'BPAN.IODF01'

Switch ID _____+

Switch configuration ID . . BASIC +

ESCON Manager lock . . 1 1. Break another user's ESCON Manager lock
2. Preserve another user's ESCON Manager lock

2. You may choose between two different kinds of switch activation:

Single switch activation

For single switch activation, use the Activate Switch Configuration panel to:

- Specify the switch ID and the switch configuration ID that is to be written to the switch.
- Select how to handle an existing ESCON Manager lock if it is in use by another user.

I/O Operations uses a locking mechanism to serialize connectivity changes across multiple users and systems. This lock allows only one user (or program) to control I/O Operations command processing at a time. If more than one user at your installation is given the ability to enter I/O Operations commands, they must synchronize their usage of I/O Operations to avoid delays or contention for I/O Operations resources.

Multiple switch activation

For multiple switch activation, use the Activate Switch Configuration panel to:

- Specify only the switch configuration ID and no Switch ID. A panel is displayed showing all switches that have a configuration under the specified name.

```

Switch Activation List
Command ==> _____ Row 1 of 2
                               Scroll ==> PAGE
Select one or more switches for activation.
Switch configuration ID . . : BASIC

/ ID Type      Description          CU    Dev
- 98 9033      First switch          0098 0098  More?
- 99 9033      Second switch         0099 0099
***** Bottom of data *****

```

Switches that are not connected to any switch device are marked with a disabled marker # in the action column, and cannot be selected for activation. The panel shows 'YES' in the 'More?' column if there are more than one control unit and one device. If there is only one control unit and one device, the 'More?' column is left blank. HCD uses the switch devices that are connected to the active system.

- Select the switches that will be activated simultaneously.

When you have finished your selection, press the Enter key. A confirm panel is displayed. On this panel you have to confirm if you want the active switch configuration of all switches shown in the list to be updated.

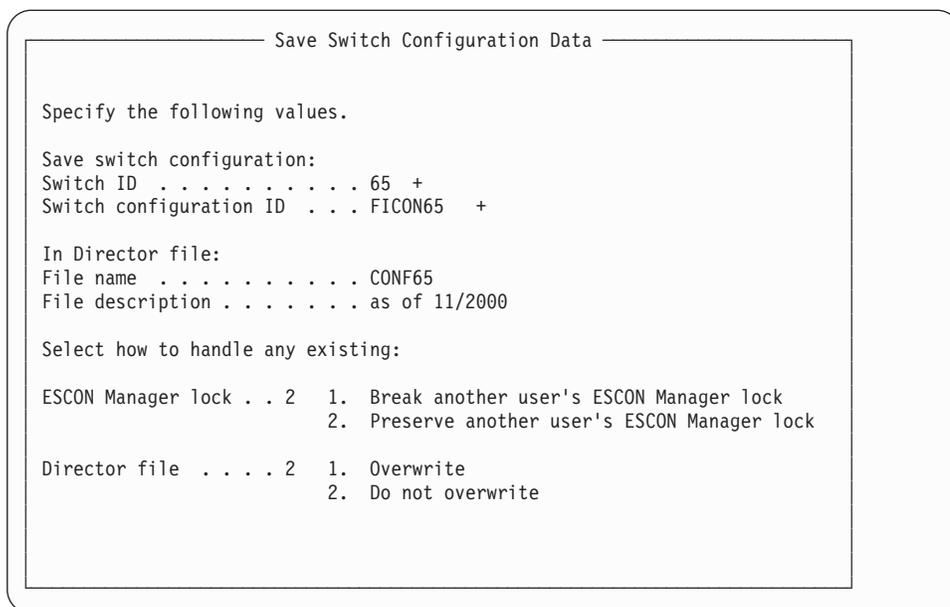
Note: In case a switch activation fails, no switch will be activated, the rule is "none or all".

For activating switch configuration data, refer to *IBM Tivoli System Automation for z/OS Planning and Installation*.

Saving switch configuration data

Use the Save switch configuration data function to save an existing switch configuration definition in a switch file. For information on prerequisites to save a switch configuration, see "Activating switch configuration data" on page 183.

1. On the *Primary Task Selection* panel, select *Activate or process configuration data*, and from the resulting panel, select *Save switch configuration*. The Save Switch Configuration Data panel is displayed.



2. Use the Save Switch Configuration Data panel to:
 - Specify the switch ID and the switch configuration ID for the configuration data that is to be saved in the switch file
 - Specify the name of the switch file used to store the switch configuration data
 - Select how to handle an existing ESCON Manager lock
 - Indicate whether to overwrite an existing switch file with the file name you specified.

Chapter 8. How to work with I/O Autoconfiguration

You can use the HCD I/O Autoconfiguration function to perform automatic configuration changes in order to define switched FICON connected or FICON directly connected DASD and tape control units and devices which are currently not yet defined in the base IODF, which can be either the active or currently accessed IODF. Proposed definitions are automatically written into a specified target work IODF which is created as a copy of the active or accessed IODF.

For I/O Autoconfiguration, HCD invokes the Input/Output Subsystem (IOS) to discover I/O hardware in the current configuration that is accessible to the system. HCD can define connections between processors and controllers through combinations of switch-attached paths and point-to-point paths. Proposed definitions are automatically written into a specified target work IODF which may be created as a copy of the active or accessed IODF. You can control the proposals for autoconfiguration changes by specifying autoconfiguration policies.

I/O Autoconfiguration is available starting with zEnterprise processors (processor type 2817). It requires the same access authorization as used for dynamic reconfiguration (see “Giving users access authority” on page 360).

During I/O Autoconfiguration processing, HCD presents the discovered controllers, control units and devices to the user and offers proposals how to configure them. The user can accept or change these definition proposals. On the user's confirmation, the configuration definitions are then written into the specified target IODF.

HCD provides a series of dialogs to perform automatic I/O configuration:

1. A dialog to define autoconfiguration policies:

Before you start the discovery processing, you can define the policies which control the automatic definition of discovered control units and devices.

For more information, refer to “How to define autoconfiguration policies” on page 188.

2. A dialog to perform the discovery and definition process:

You define the scope of discovery, select the autoconfiguration operation mode (attended or unattended fast-path mode), select the IODF against which the discovery should run and the target IODF that receives the resulting configuration definitions for all discovered new or changed controllers.

After having defined or revised the mentioned options, you can invoke the discovery and definition processing. Depending on the selected operation mode, the dialog either leads you through the configuration steps or directly defines the discovered controllers and devices in the target IODF.

For more information, refer to “How to perform automatic I/O configuration” on page 196.

Through determining the configuration characteristics of the discovered control unit as established at the controller side, and through establishing a recommended configuration based on availability and performance considerations, this functionality ensures that the controller configuration matches its logical definition.

After a successful run of I/O Autoconfiguration, you have complete update control regarding the I/O definitions written to the target IODF.

How to define autoconfiguration policies

Before you let HCD discover and define control units and I/O devices, you must specify your desired autoconfiguration policies. This task comprises the following subtasks:

- “How to set keywords for autoconfiguration policies”
- “How to define logical partition groups for autoconfiguration” on page 193
- “How to define OS groups for autoconfiguration” on page 194

How to set keywords for autoconfiguration policies

To set the autoconfiguration policies by means of keywords, perform the following:

- Select *Edit profile options and policies* from the HCD *Primary Task Selection* panel to invoke the *Profile Options and Policies* menu (Figure 84).

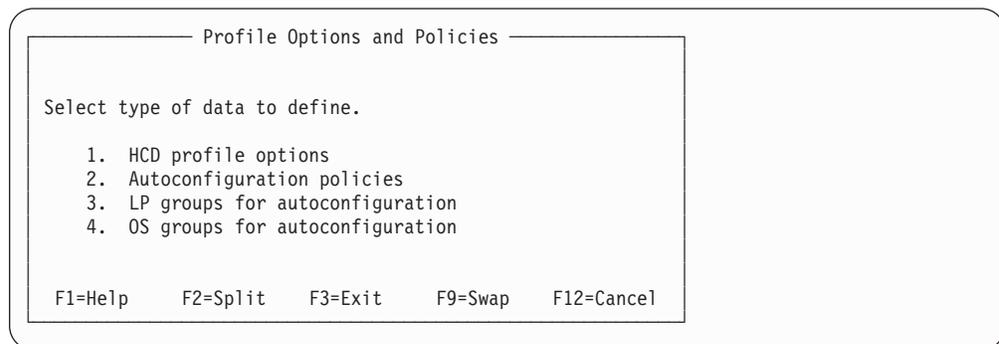


Figure 84. Profile Options and Policies

- Select **Autoconfiguration policies**.
- This invokes the *Autoconfiguration Policies* dialog which displays the current value settings for autoconfiguration policy keywords as they are either explicitly set in the HCD profile data set or as they are defaulted by HCD. Use this dialog to revise or change the displayed keyword values.

The autoconfiguration policies are saved in the HCD profile (see also “Defining an HCD profile” on page 19 for information on how to edit the HCD profile data set).

```

Autoconfiguration Policies
Row 1 of 13 More: >
Command ==> _____ Scroll ==> CSR

Edit or revise autoconfiguration policies.

HCD Profile : DOCU.HCD.PROFILE

/ Policy keyword      P Value +
_ AUTO_CHPID_EXCLUDE  N SYSA.0,00-09
_ AUTO_CHPID_INCLUDE  N SYSA.1,1A-1F
# AUTO_MATCH_CU_DEVNUM Y YES
# AUTO_SS_ALTERNATE   Y 1
# AUTO_SS_DEVNUM_SCHEME Y PAIRING
# AUTO_SUG_CU_RANGE   Y 0001-FFFE
# AUTO_SUG_DEV_RANGE  Y 0001-FFFF
# AUTO_SUG_DYN_CHPIDS Y 2
# AUTO_SUG_LPGROUP    N _____
# AUTO_SUG_OSGROUP    N _____
# AUTO_SUG_STAT_CHPIDS Y 6
_ AUTO_SWAD_EXCLUDE   N 1A
_ AUTO_SWAD_INCLUDE    N _____
***** Bottom of data *****

```

Figure 85. Autoconfiguration Policies

Column *P* is set to Y (for yes), if a change of the policy key value is active immediately. Value N denotes that the value change only becomes active with a new discovery. For more information, refer to “How to change autoconfiguration policies during the autoconfiguration process” on page 192.

You can scroll to the right in this list to see a *Description* column for each policy keyword. This column may display an existing user comment, which you can change, or you can specify a new comment.

For keywords that you may specify multiple times with different values, for example, AUTO_CHPID_EXCLUDE, actions *Add* and *Delete* are available from the context menu.

To retrieve an online explanation of a keyword, move the cursor into its *Value* column and press the F1 key.

The subsequent topics list and explain the available keywords for defining your desired autoconfiguration policy.

Exclude CHPIDs from the discovery

AUTO_CHPID_EXCLUDE

This policy specifies a single CHPID number or a range of CHPID numbers that are excluded from being used for discovery or channel path assignment for a specific channel subsystem (by giving its ID) or all channel subsystems (by specifying *) of a certain processor. The keyword can occur multiple times.

The following example specifies that CHPIDs 04, 20 to 2F, and 42 are not used for channel subsystem PROCA.0. CHPID 42 is also not used for all other channel subsystems of processor PROCA.

```

AUTO_CHPID_EXCLUDE = PROCA.0,04
AUTO_CHPID_EXCLUDE = PROCA.0,20-2F
AUTO_CHPID_EXCLUDE = PROCA.*,42

```

Include CHPIDs for the discovery

AUTO_CHPID_INCLUDE

This policy specifies a single CHPID number or a range of CHPID numbers that should be considered for discovery and channel path assignment for a specific channel subsystem (by giving its ID) or all channel subsystems (by specifying *) of a certain processor. The keyword can occur multiple times. If this option is specified, no other channel paths are considered.

The following example specifies that CHPIDs with numbers 04, 20 - 2F, and 42 are used for discovery and definition from channel subsystem PROCA.0. CHPID 42 may also be used from all other channel subsystems of processor PROCA. No other unspecified CHPID number for processor PROCA is considered for discovery and definition.

```
AUTO_CHPID_INCLUDE = PROCA.0,04
AUTO_CHPID_INCLUDE = PROCA.0,20-2F
AUTO_CHPID_INCLUDE = PROCA.*,42
```

If no value is specified for a given channel subsystem, all CHPID numbers (00 - FF) are considered for discovery and channel path assignment.

Control unit number should match base device number

AUTO_MATCH_CU_DEVNUM

This policy specifies whether for autoconfiguration definitions a control unit number should match the starting base device number.

If you specify YES (which is the default), the first base device is set to the same number as the control unit. If NO is specified, the device number of the first base device and the control unit number do not necessarily need to match.

Alternate subchannel set for proposed alias devices

AUTO_SS_ALTERNATE

This policy specifies the ID of the subchannel set in which newly discovered PAV alias devices are defined during an auto-definition process, provided that free device numbers are available in this subchannel set, and processors that have access to the device range, support alternate subchannels.

The default subchannel set ID is 1.

Subchannel set device numbering scheme

AUTO_SS_DEVNUM_SCHEME

This policy defines the scheme for assigning device numbers to PAV alias devices in an alternate subchannel set.

Supported schemes are:

CONSECUTIVE

The alias device numbers in an alternate subchannel set are consecutive to the base device numbers.

DENSE

The device numbers in an alternate subchannel set are densely assigned, that is, the next free device numbers in the assigned device number range are used.

PAIRING

Base and alias device numbers are assigned alternatively starting

with, for example, device numbers xx00 and xx80 for base devices versus xx80 and xx00 for alias devices.

NONE

Device and control unit numbers are not automatically applied. Instead, HCD presents the new discovered control units with a control unit number 0000 in the *Proposed Control Unit List* and the new devices with starting device number 0000 in the subsequent *Proposed Control Unit / Device List*. You can thus manually insert free control unit numbers, respectively device numbers on the according lists, which are validated by HCD. HCD issues a message to notify you to assign the numbers.

If you work in attended operation mode, (that is, Show proposed definitions has been set to Yes in Figure 86 on page 197), HCD once more offers you the opportunity to review your input by redisplaying both lists with the updated numbers after you pressed enter. All actions are now available on the lists as in any other setting of the AUTO_SS_DEVNUM_SCHEME policy.

Otherwise, if you work in unattended fast-path operation mode, HCD accepts the numbers you entered and processing proceeds.

PAIRING is the default.

Control unit number range for auto proposal

AUTO_SUG_CU_RANGE

This policy specifies the range of control unit numbers from which numbers for auto-defined control units are taken. If no value is specified, range 0001-FFFF is taken as default.

Specify the range according to the following syntax:

nnnn-mmmm, where: nnnn is the lower range boundary,
mmmm is the upper range boundary.

Device number range for auto proposal

AUTO_SUG_DEV_RANGE

This policy specifies the range of device numbers from which device numbers for auto-defined devices are taken. If no value is specified, range 0001-FFFF is taken as default.

Note: For *AUTO_SUG_DEV_RANGE* and *AUTO_SUG_CU_RANGE*, I/O Autoconfiguration avoids using device and control unit numbers in the 0000-00FF range in subchannel set 0. If you have no alternative number ranges available, you must configure the CUs or devices manually in this range.

Number of static CHPIDs to be assigned

AUTO_SUG_STAT_CHPIDS

This policy specifies the number of static channel paths to be assigned to a control unit definition, if it is auto-defined. At least one and not more than 8 static channel paths can be defined.

The default is 6.

Maximum number of dynamic CHPIDs

AUTO_SUG_DYN_CHPIDS

This policy specifies the number of dynamically managed channel paths

allowed on a control unit definition, if it is auto-defined. A maximum number of 7 dynamic channel paths is allowed, however, the sum of `AUTO_SUG_STAT_CHPID`s and `AUTO_SUG_DYN_CHPID`s must not exceed 8.

The default is 2.

LP group for autoconfiguration

AUTO_SUG_LPGROUP

This policy specifies the name of a group of logical partitions to which discovered devices are assigned. If no name is set, devices are assigned to all partitions of the active sysplex. The reserved group name `ALL` signals this during autoconfiguration processing.

OS group for autoconfiguration

AUTO_SUG_OSGROUP

This policy specifies the name of a group of OS configurations to which discovered devices are assigned. If no name is set, devices are assigned to all OS configurations which correspond to the active LP group.

Exclude switch addresses from the discovery

AUTO_SWAD_EXCLUDE

This policy specifies a single switch address or a range of switch addresses that are excluded from being used for discovery or channel path assignment. Note that switch addresses have to be specified rather than switch IDs. The keyword can occur multiple times.

The following example specifies that switches with addresses 14 and 20 to 2F are not used for discovery and channel path assignments:

```
AUTO_SWAD_EXCLUDE = 14
AUTO_SWAD_EXCLUDE = 20-2F
```

If no value is specified, no switch is excluded from discovery and channel path proposal.

Include switch addresses for the discovery

AUTO_SWAD_INCLUDE

This policy specifies a single switch address or a range of switch addresses that can be used for discovery and channel path assignment. Note that switch addresses have to be specified rather than switch IDs. The keyword can occur multiple times.

The following example specifies that switches with addresses 14 and 20 to 2F can be used for discovery and channel path assignments. No other unspecified switch address is considered for discovery and channel path assignments.

```
AUTO_SWAD_INCLUDE = 14
AUTO_SWAD_INCLUDE = 20-2F
```

If no value is specified, all switch addresses in the range 00 to FF can be used for discovery and channel path proposal.

How to change autoconfiguration policies during the autoconfiguration process

You can change your defined autoconfiguration policies between two subsequent controller discoveries without the need to restart I/O Autoconfiguration. For this

purpose, in the *Discovered New or Changed Controller List* dialog (Figure 87 on page 199), select the *Policy* action bar choice, and then select **1. Change policy options** to invoke the *Autoconfiguration Policies* dialog as described in “How to set keywords for autoconfiguration policies” on page 188. You can now change the values of certain keywords that you want to apply on the subsequent controller discovery and control unit autoconfiguration. However, only changes for those keywords become effective immediately for which HCD sets column *P* to Y in the *Autoconfiguration Policies*, for example, AUTO_MATCH_CU_DEVNUM (see also Figure 85 on page 189).

Changes of all policy keywords denoted with value N in column *P* are also possible between two controller discoveries, but require a new fabric discovery (as described in “How to perform automatic I/O configuration” on page 196) to become effective.

How to define logical partition groups for autoconfiguration

A logical partition group (LP group) is a collection of logical partitions containing z/OS systems that belong to the same sysplex. This collection is used by I/O Autoconfiguration to determine to which partitions the discovered devices should be assigned.

Selecting option **3 LP groups for autoconfiguration** from the *Profile Options and Policies* menu (see Figure 84 on page 188) invokes the *Autoconfiguration LP Group List* which displays a list of partition groups (LP groups).

Use the *Autoconfiguration LP Group List* to define or delete LP groups, to assign logical partitions to a group or unassign partitions from a group.

```

Autoconfiguration LP Group List
Row 1 of 2
Command ==> _____ Scroll ==> CSR

To view assigned partitions, select one or more LP groups,
then press Enter. To add an LP group, use F11.

/ LP group name  Description
- LPGROUP0      COM1
- LPGROUPX      COM1, SYSA1
- LPGROUP0      SYSA1, SYSA2, SYSA3, SYSA4
- LPGROUP2      SYSA1, SYSA2
- LPLGROUP      SYSA1, SYSA3, SYSA4
***** Bottom of data *****

F1=Help      F2=Split      F3=Exit      F4=Prompt
F7=Backward  F8=Forward    F9=Swap      F11=Add
F12=Cancel   F22=Command

```

Invoking action *Add like* with action code **a** for an LP group, or just pressing F11, invokes the *Add Autoconfiguration LP Group* dialog with two entry fields for the LP group name and an optional user description.

Invoking action *View/Assign logical partitions* with action code **s** for an LP group invokes the *Autoconfiguration LP Group Assignment List*. This list displays all logical partitions assigned to the named LP group. You can assign a new partition to or unassign an included partition from the LP group.

```

Autoconfiguration LP Group Assignment List
Row 1 of 2
Command ==> _____ Scroll ==> PAGE

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

LP group name : LPGROUP0      SYSA1, SYSA2, SYSA3, SYSA4

/ Partition Name      Description
- P2817.0.SYSA1      logical Partition 1, CSS0
- P2817.1.SYSA2      logical Partition 2, CSS1
- P2817.2.SYSA3      logical Partition 3, CSS2
- P2817.2.SYSA4      logical Partition 4, CSS2
***** Bottom of data *****

F1=Help      F2=Split      F3=Exit      F4=Prompt
F7=Backward  F8=Forward    F9=Swap      F11=Add
F12=Cancel   F22=Command

```

To add a partition to an LP group, press F11 in this list to invoke the *Add Partition to LP Group* dialog.

```

Add Partition to LP Group

Specify the following values.

LP group name . . . . . : LPGROUP0      SYSA1, SYSA2, SYSA3, SYSA4

Processor ID . . . . . : P2817__ +
Partition Name . . . . . : SYSA5__ +

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

```

In the entry fields *Processor ID* and *Partition Name*, specify the name of the processor and the name of the partition which you want to add to the LP group. After you entered a processor name, the dialog lets you select a certain partition from that processor. You can only enter processor IDs from processors that are defined in the currently accessed IODF.

How to define OS groups for autoconfiguration

An OS group is a collection of OS configurations which is used by autoconfiguration to determine to which operating systems of type MVS the auto-defined devices should be assigned.

Selecting option 4 **OS groups for autoconfiguration** from the *Profile Options and Policies* menu (see Figure 84 on page 188) invokes the *Autoconfiguration OS Group List* which displays a list of operating system groups.

Use the *Autoconfiguration OS Group List* to view, add, or delete OS groups or to assign autoconfigured devices to operating systems.

```

----- Autoconfiguration OS Group List -----
                                         Row 1 of 1
Command ==> _____ Scroll ==> CSR

To view assigned OS configurations, select one or more OS
groups, then press Enter. To add an OS group, use F11.

/ OS group  Description
_ OSGROUP1  Currently active OS group
***** Bottom of data *****

F1=Help      F2=Split      F3=Exit      F4=Prompt
F7=Backward  F8=Forward    F9=Swap      F11=Add
F12=Cancel   F22=Command

```

Invoking action *Assign operating system config* with action code **S** for an OS group invokes the *Autoconfiguration OS Group Assignment List*. This list displays all operating system configurations assigned to the named OS group. You can assign a new operating system to or unassign an included operating system from the OS group.

```

----- Autoconfiguration OS Group Assignment List -----
                                         Row 1 of 1
Command ==> _____ Scroll ==> CSR

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

OS group name : OSGROUP1      Currently active OS group

/ OS Configuration ID  Description
_ OS1                  Primary OS
_ OS2                  MVS Test System
_ OS3                  Backup
***** Bottom of data *****

```

To add an OS configuration to an OS group, press F11 in this list to invoke the *Add Operating System Configuration to OS Group* dialog.

```

----- Add Operating System Configuration to OS Group -----

Specify the following values.

OS group name . . . . : OSGROUP1      Currently active OS group
OS configuration ID . . OS4_____ +

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

```

For the entry field *OS configuration ID*, this dialog provides prompting support to let you easily select a certain OS configuration defined for the sysplex.

How to perform automatic I/O configuration

After having specified all policies as described in “How to define autoconfiguration policies” on page 188, you can let HCD try to discover and automatically define control units and I/O devices into a specified target IODF.

The I/O autoconfiguration process consists of the following steps:

1. *The fabric discovery:* You invoke the I/O Autoconfiguration function from the HCD *Primary Task Selection* panel. This in turn causes HCD to invoke IOS to perform the fabric discovery process. You can select the scope of discovery by searching all controllers, new controllers only, search for the controller containing a specific control unit, or search for a controller with a certain serial number or for multiple controllers matching a serial number pattern. The result of the fabric discovery is the list of controllers that is new or has changed compared to the base IODF.

Note that the scope of discovery is the active sysplex.

2. *The controller discovery:* From the discovered controllers, HCD retrieves and proposes control unit and device types and numbers, channel path assignments, partition access, and OS device parameters. You can choose whether HCD should perform the definition without user interaction, or whether the dialog should show the proposed definitions so that you can confirm or change these values.

I/O Autoconfiguration makes temporary changes to the active I/O configuration by adding devices that are used exclusively for discovery on the targeted systems in order to search for attached devices.

You invoke the I/O Autoconfiguration process from the HCD dialog as follows:

1. From the *Primary Task Selection* panel select *1. Define, modify, or view configuration data*.
2. Then select *6. Discovered new and changed control units and I/O devices*.

HCD invokes the *Discovery and Autoconfiguration Options* dialog shown in Figure 86 on page 197 which lets you select processing options for discovery and autoconfiguration.

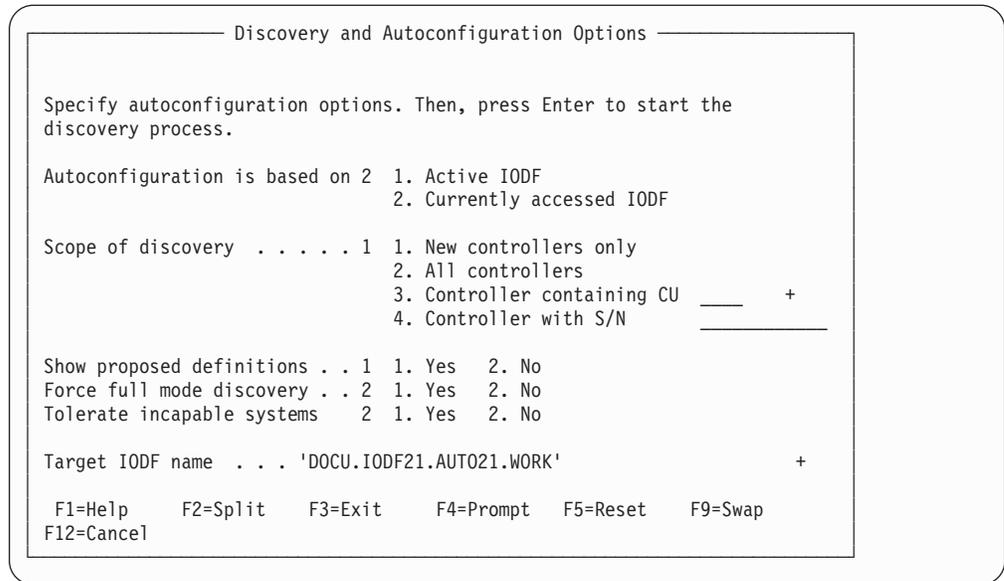


Figure 86. Discovery and Autoconfiguration Options

Specify your desired processing options for discovery and autoconfiguration definition:

Autoconfiguration is based on

Choose whether the active production IODF or the currently accessed IODF should be taken as the base for new configuration definitions resulting from the discovery process.

This means, HCD checks the discovered devices accessible to the system against this selected IODF, whether they are already defined or not. If they are found to be new or changed, the resulting configuration proposals are also adopted to fit into this IODF.

HCD copies this IODF to the selected target IODF (see *Target IODF name* later in this list) which receives all changes done to the configuration during autoconfiguration processing.

Note: You can use an IODF as base IODF, if it represents the active configuration or a potential target configuration for the next dynamic activation.

Scope of discovery

With this option you decide about the controllers to be discovered:

- **New controllers only:** HCD discovers and returns only new controllers, which are not yet known to the LPARs in the selected LP group.
- **All controllers:** HCD discovers and returns all new controllers as well as all changed controllers.
- **Controller containing CU:** HCD performs a discovery limited to that controller containing the control unit with the specified number. The referenced control unit must be a DASD or tape control unit and must be defined in the base IODF.
- **Controller with S/N:** HCD performs a discovery limited on that controller with the specified serial number or on 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. A wildcard matches any number of characters. For example, F30WD, F3*, *F3*, *WD, or *WD* are valid values to include controller F30WD.

Show proposed definitions

You can decide whether the dialog should display proposed definitions for possible configuration changes. Select Yes if you want to work in an attended operation mode. In this mode, HCD invokes a subsequent series of dialogs in which you can revise and change the proposed settings. How to work in this attended mode is described in “How to apply updates to the autoconfiguration proposals” on page 199.

Select No if you want to run the unattended fast-path of I/O Autoconfiguration. In this case, HCD does not offer a possibility to revise the proposals or to update or add definitions. Instead, the HCD definitions are completely saved in the target IODF immediately. However, if you specified AUTO_SS_DEVNUM_SCHEME = NONE in your autoconfiguration policy, the dialogs where you can define control unit and device numbers are shown.

Force full mode discovery

Decide when discovery processing should stop. If set to No, which is the default, processing stops after several consecutive unused CUADD values that do not exist on a target controller. With this option set to Yes, for each discovered controller, all unused logical control unit addresses (CUADD values) and unit addresses are checked for changes.

Tolerate incapable systems

Decide whether discovery processing should stop if the discovery target scope (specified with policy AUTO_SUG_LPGROUP)

- includes partitions that contain systems that are not able to perform I/O autoconfiguration because of missing support in software or hardware, or
- if it includes partitions that are not active or that are containing systems that do not belong to the current sysplex.

If set to No, which is the default, processing stops if the active LP group contains an inactive system or a system not capable for autoconfiguration. With this option set to Yes, incapable or inactive systems are ignored and excluded from the active LP group.

Target IODF name

Type the name of a work IODF that will receive the configuration definitions for all discovered new or changed controllers, according to your selected scope of discovery.

This input is required. The specified IODF can either be an existing work IODF, or it is created by HCD. In any case, the IODF specified in the *Autoconfiguration is based on* entry field is copied to the specified target IODF.

The target IODF must not be enabled for multi-user access.

Note: As soon as you accepted any proposals into your target IODF, it becomes the new currently accessed IODF.

How to apply updates to the autoconfiguration proposals

This topic explains how to work in the attended operation mode, that is, with option *Show proposed definitions* set to 1 (Yes) in the **Discovery and Autoconfiguration Options** panel shown in Figure 86 on page 197.

“How to work in unattended mode” on page 203 explains the unattended fast-path discovery.

After specifying your desired options on the **Discovery and Autoconfiguration Options** panel, pressing Enter starts the discovery process. HCD notifies users with the message: *FABRIC discovery in progress - please wait ...*

After a successful discovery, HCD displays the result in the *Discovered New or Changed Controller List* (Figure 87). Only discovered controllers are displayed, which are reachable from all target systems, which are capable of discovery, and which have partitions defined in the LP group referenced by the AUTO_SUG_LPGROUP policy.

/	Type	Model	Name	Plant	Serial-#	New	Processed
-	2107	9A2	IBM	13	15663	Yes	No
-	2107	921	IBM	13	34211	Yes	No
-	2107	922	IBM	13	67884	Yes	No
-	3491	A01	IBM	13	18321	Yes	No
7	1750	511	IBM	13	28824	No	No
-	1750	522	IBM	13	43560	No	No
-	2105	F10	IBM	13	28251	No	No
-	2105	F20	IBM	13	12353	No	No
-	2105	800	IBM	13	17791	No	No

Figure 87. *Discovered New or Changed Controller List*

This panel lists all discovered controllers which are either not yet defined in the IODF, or whose definition in the IODF is different from discovered controller characteristics.

If you selected AUTO_SS_DEVNUM_SCHEME = NONE, all control unit numbers for new controls are prefilled with 0000. You see message *Assign numbers for control units or devices. Then hit Enter*. Exclude all control units which should not be added, and assign control unit numbers. Pressing Enter processes the line commands or edited fields. After pressing Enter a second time with no changes on the list, you see the message *Items have been processed. Review them, then press Enter*. You are then in the normal process flow as if control unit numbers have been proposed.

On this dialog, using action code **F7**, you can select multiple controllers that you want to be defined or changed in the target IODF. HCD subsequently processes each selected controller in the way described in the remainder of this topic.

Pressing Enter on this panel with selected controllers starts the controller discovery and definition process for the next selected controller. Users are notified with the message: *CONTROLLER discovery in progress - please wait ...* As a result of the discovery process, the *Proposed Control Unit List* offers definition proposals for the control units found in the currently processed controller.

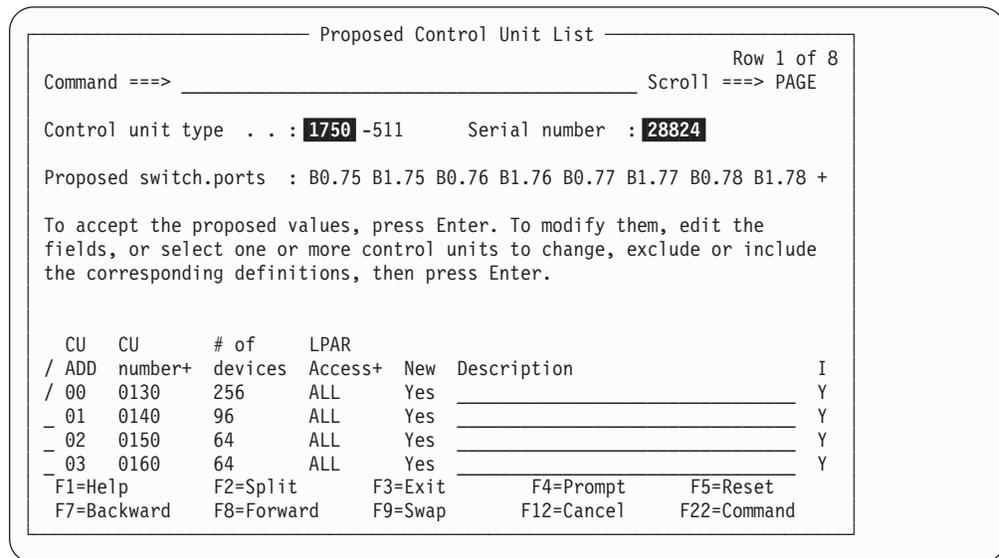


Figure 88. Proposed Control Unit List

The panel shown in Figure 88 shows proposed definition details for existing or new control units configured in the discovered controller with the serial number **28824** displayed in the header of the panel. The indicated controller type **1750** has been determined by discovery, and is the proposed control unit type for all listed control units within the controller.

All control units are connected to the proposed switch ports as listed in the upper part of the panel. A plus sign (+) at the end of the switch port list signals, that the control units will have connections to more than eight switch ports. You can see the full list of connected switch ports for a control unit on the *View Control Unit Definition* dialog.

You can accept the proposed control unit definitions, or you can perform the following modifications:

- For control units showing Yes in column *New* (which indicates whether the control unit is not yet defined in the IODF, or is not yet connected to any of the LPARs defined in the AUTO_SUG_LPGROUP policy), you can overwrite the values in column *CU number*.
- Also, you can overwrite the values in the *LPAR Access* and *Description* fields in this dialog. Note that when changing the prefilled LP group name in the *LPAR Access* column, you can only replace it by the name of another LP group, which defines a subset of those logical partitions contained in the LP group used during the autoconfiguration propose step.
- With action code **F7** you get an overview of available actions on the selected control units:

- With action code **i**, you can include the corresponding control unit definition into the target IODF, or with action code **e**, you can exclude the control unit from being defined. Your selection is reflected in column *I*: Y denotes included and N denotes excluded control units.
- Using action code **c** leads you to the *Select Processor / CU* panel (see Figure 56 on page 128). On this window, HCD displays a list of all defined processors. You can define how the control unit is to be attached to one or more processors.

When the controller discovery returns control units and devices, HCD checks the discovered switch ports (Figure 88 on page 200) for existing definitions in the target IODF to match the existing control unit/device numbers with the proposed ones and proceeds like follows:

Rules for discovered control units:

- For each discovered control unit that is already defined with the same CUADD value, the existing control unit definition is checked for the same serial number. If the serial numbers match, or the IODF definition does not contain a serial number, the control unit number of the existing control unit is used and updated with the serial number of the discovered control unit. If the serial numbers do not match, a warning message is given, and the discovered control unit is proposed to be newly defined with a new serial number. It is, however, recommended that the serial numbers match, or the serial number of the defined control unit should be blank.
- For each discovered control unit that is not yet defined in the IODF, a new control unit number is proposed.
- When a new control unit number is proposed, its value is taken from the preferred range specified by policy `AUTO_SUG_CU_RANGE`. If there is no free control unit number in the IODF within that range, a warning message indicates that the policy could not be followed, and a free control unit number outside of the range is proposed.
- Proposed existing control units are updated with the discovered serial number. If the type of a discovered control unit differs from its definition in the IODF, the definition is updated.
- Some controllers may respond indicating over-defined unit address configurations. In these situations, devices are included in the proposal for unit addresses that are not configured on the controller. You can leave these devices in the configuration, or you can remove them from the proposal.

Rules for discovered devices:

- For each discovered device that is already defined with the same unit address on an existing control unit, the existing device number is proposed. In that case, the definition may differ from the specified policy, for example for the subchannel set number.
For non-existing devices on the control unit, the existing device numbering scheme is applied if possible.
- For new devices on new or existing control units where the existing device number scheme could not be applied, the device numbers are determined based on policies `AUTO_SUG_DEV_RANGE` and `AUTO_MATCH_CU_DEVNUM`. For PAV alias devices the numbers are additionally determined based on policies `AUTO_SS_ALTERNATE` and `AUTO_SS_DEVNUM_SCHEME`. If a policy could

not be applied because no free numbers are available for the active LP group and OS groups, a warning message is given and free device numbers outside the policies may be used.

You can now apply desired modifications and press Enter after you are finished, or you can accept the proposed definitions without changes and press Enter. In both cases, HCD now displays the *Proposed Control Unit / Device List*.

```

Proposed Control Unit / Device List                               Row 1 of 8
Command ==> _____ Scroll ==> PAGE
Control unit type . . : 1750-511      Serial number : 28824

To accept the proposed values, press Enter. To modify them, edit the
fields, or select one or more device ranges to change, exclude or include
the corresponding definitions, then press Enter.

-----Device----- S CU  UA  OS
/ Number  Type+  S Num Range Access+  N Description      I
- 6300,128 3390B  0 0130 00-7F ALL    N _____      Y
- 6380,128 3390A  1 0130 80-FF ALL    N _____      Y
- 6400,32  3390B  0 0140 00-1F ALL    N _____      Y
- 6420,64  3390A  1 0140 20-5F ALL    N _____      Y
- 6500,16  3390B  0 0150 00-0F ALL    Y _____      Y
- 6510,48  3390A  1 0150 10-3F ALL    Y _____      Y
- 6600,16  3390B  0 0160 00-0F ALL    Y _____      Y
- 6610,48  3390A  1 0160 10-3F ALL    Y _____      Y
***** Bottom of data *****

```

Figure 89. Proposed Control Unit / Device List

This list proposes definition details for existing or new devices accessible by the currently processed discovered control units (0130 - 0160 from Figure 89). In the header of this panel, you can see the control unit type and serial number of the discovered controller.

If you selected `AUTO_SS_DEVNUM_SCHEME = NONE`, all device numbers for new devices are prefilled with 0000. You see a message *Assign numbers for control units or devices. Then hit Enter*. Exclude all devices which should not be added, and assign device numbers. Pressing Enter processes the line commands or edited fields. After pressing Enter a second time with no changes on the list, you see the message *Items have been processed. Review them, then press Enter*. You are then in the normal process flow as if device numbers have been proposed.

You can accept the proposed device definitions without changes by pressing Enter. Also, you can narrow by overtyping one or more of the device ranges, but only those with a Y in column N (abbreviation for New), which indicates that the device range is not yet defined in the IODF.

Furthermore, for one or more of the listed device ranges with Y in column N, you can change the *OS Access* and the *Description* fields by overtyping the values in the panel. Again, a changed OS group must be a subset of the initial OS group.

For further available actions on devices, select one or more devices using action code **7**:

- By selecting action code **i**, you can include, or with action code **e**, you can exclude the corresponding devices from autoconfiguration.
- Using action code **c** leads you to the *Device / Processor Definition* panel (see Figure 62 on page 137). On this panel, HCD displays a list of all defined

processors that have one or more channel paths to the control unit to which the device being added or changed is attached. Here you can select the processor/CSS(s) for which you want to change the device-to-processor definition.

In both cases, either with modifications applied or with accepting the unchanged proposals, pressing Enter lets you return to the *Discovered New or Changed Controller List*. For each successfully handled controller, its *Processed* field is now turned to Yes (see Figure 87 on page 199).

You can select the next controller for being auto-configured, or you can exit the dialog. Exiting the dialog saves all applied configuration changes in the target IODF.

For documenting the progress of your configuration actions, and for later reference, you can save the lists from Figure 87 on page 199, Figure 88 on page 200, and Figure 89 on page 202 in a data set using the SAVE command as described in “How to print list panels” on page 263.

How to work in unattended mode

To run the unattended fast-path mode for I/O Autoconfiguration, specify 2 (No) for option *Show proposed definitions* from the *Discovery and Autoconfiguration Options* dialog shown in Figure 86 on page 197. In this case, after pressing Enter to start the discovery process, HCD notifies the user on the *Discovered New or Changed Controller List* that the discovery process is started: *CONTROLLER discovery in progress - please wait* After a successful autoconfiguration run, with all proposals automatically defined without user interaction, HCD sets the *Processed* field in Figure 87 on page 199 to Yes for each successfully configured controller. However, if you specified policy `AUTO_SS_DEVNUM_SCHEME = NONE`, I/O Autoconfiguration asks you to enter your own control unit and device numbers as described in “How to apply updates to the autoconfiguration proposals” on page 199.

Prerequisites and operational considerations for using I/O Autoconfiguration

Consider the following information when exploiting the I/O Autoconfiguration functionality of HCD:

- The I/O Autoconfiguration process requires that the systems in the partitions from the LP groups are running on a zEnterprise 196 (z196) processor with at least z/OS V1R12.
- The target work IODF should be equal to, or be a descendent of the active IODF. This restriction is not enforced but recommended to facilitate consistent discovery of devices. If devices need to be added to do discovery, failures may occur due to inconsistent IODFs. Hardware definitions of this active IODF will remain consistent.
- All active IODFs for the systems in a SYSPLEX should be the same. This restriction is not enforced, but allows HCD to optimize the discovery process for the different systems by reusing the defined configuration. Tokens should be in sync, prior activates should have been completed.
- Without indicating *Force full mode discovery*, there is a limit on the number of subsequent failures. Unless force full mode discovery is requested, processing

assumes that CUADD values start at 00 and continue to the last defined value in consecutive order, with no missing CUADDs.

- For DASD controllers, all newly discovered devices are assumed to be 3390 type devices, either type 3390A or 3390B.
- Switches may have port restrictions via zoning or via using a switch matrix, that limit the ability of a CHPID to connect to a destination port of a control unit interface. If such port restrictions exist within a switch, I/O Autoconfiguration may configure paths that cannot be used. If a port is discovered on a controller, it is assumed that it has access to all configured logical control units on that controller.
- Only accessible FICON CHPIDs, switches, and ports that are configured online are considered during discovery. CHPIDs connected to switches, must be fully defined and include actual values in the Dynamic entry swith ID, Entry swith ID, and Entry port fields.
- At least one system per CPC must have the ability to perform dynamic I/O configuration changes. This system needs not be part of the target LP group.
- A logical control unit containing only secondary devices in an active PPRC relationship may not be able to be discovered. The I/O processing used to determine the devices configured on a logical control unit cannot be performed on secondary devices.
- I/O Autoconfiguration is a configuration tool that configures for availability. You can use Dynamic CHPID Management (DCM) for performance management. CHPID and path selection of I/O Autoconfiguration minimizes or even eliminates single points of failure for newly discovered logical control units. DCM manages the performance by adding CHPIDs and paths to the logical control units as needed.
- Within a target LP group, I/O Autoconfiguration proposes definitions only for controllers that are consistently defined or absent for the target LP group systems in the base IODF. If a controller is partially defined in the LP group, meaning that some systems have logical control units and devices configured that others do not have, I/O Autoconfiguration does not propose definitions for the systems within the LP group that do not have the control units. In such a situation, you can control the target discovery scope using LP groups that contain only systems which require the definition.
- If candidate access lists currently exclude an LPAR from accessing a control unit already defined on a CSS, I/O Autoconfiguration does not discover and add that control unit. Therefore, it is recommended that all systems in the participating LP groups should have a homogeneous view of the devices and control units. If this is not the case, you can update device candidate lists in HCD to add devices and control units to the desired LPARs before you start I/O Autoconfiguration.
- If switches are connected such that it would be possible to have three or more switches in a path to a control unit, it is possible that this path would be chosen if no viable alternative exists.
- HCD requires that either all or none of the switches and ports in the path from the CHPID to the control unit are defined in the target IODF. Otherwise, path validation may report errors.
- Discovery attempts should be performed during times where changes are minimal. ACTIVATE processing and CF CHP commands may affect discovery processing and should be avoided as far as possible.
- I/O Autoconfiguration can write diagnostic messages to the SYSLOG to help understand the processing decisions. To enable this, add the TRAPS NAME(IOSZDACMSG) statement to the DIAGxx member that is currently in

use and then issue the SET DIAG=xx command to refresh the current settings. See *z/OS MVS Initialization and Tuning Reference* for information on the TRAPS NAME() keyword.

- When performing autoconfiguration for a processor, consider including LPARs on all CSSs on the processors in the discovery scope (using the AUTO_SUG_LPGROUP option). Note that if only a single CSS is requested in the scope of the discovery and autoconfiguration option, subsequent attempts to discover and autoconfigure on other CSSs for that same processor may experience *resources exceeded* conditions when spanned CHPIDs are in use. If the discovery must use the scope of a single CSS, discovered control units and devices may need to be manually copied to the other CSSs using HCD panels.
- Several devices are required for the discovery process. The active configuration must have devices available for discovery for all target systems. Requirements include a free range of 256 consecutive device numbers to be used for disk exploration, a free range of 16 consecutive device numbers to be used for tape exploration, and another single free device to be used to explore the fabric for controllers.
- During a discovery process, avoid removing or adding systems to the sysplex. This could cause the active discovery attempt to fail. If you need to IPL or remove a system in the sysplex, you can exit the I/O Autoconfiguration process and resume it once the systems have been IPLed or removed.
- If a discovery would produce a proposal containing a two-byte link address for a control unit, hinting to a switch connection for this control unit, but no switch definition is found in the IODF, then the proposal fails. HCD issues a message containing the missing switch address. Users must now first define the switch using the reported switch address and then repeat the discovery. HCD then performs the control unit's connection to the switch automatically.
- If a proposal for a control unit contains a switch port that is defined as uninstalled or occupied in the IODF, HCD automatically changes the switch port to either installed or not occupied and then defines the connection.
- When a mixture of switched and directly attached paths to a control unit are found, path proposal processing only creates a set of all switched paths or all directly attached paths. Path proposal processing favors directly attached paths over switched paths in most cases, except for when a set of switched paths can satisfy the number of requested static paths when not enough directly attached paths are available for proposal.

Chapter 9. How to activate or process configuration data

Overview

This information unit describes how to:

- “Build a production IODF” on page 208
- “Build an IOCDS” on page 212
- “Build z Systems cluster IOCDSs” on page 215
- “Manage z Systems cluster IPL attributes” on page 218
- “Build an IOCP input data set” on page 219
- “Create JES3 initialization stream checker data” on page 226
- “Build I/O configuration data” on page 226
- “Verify an I/O configuration” on page 227
- “Activate a configuration dynamically” on page 230
- “Activate a configuration sysplex-wide” on page 234
- “Activate a configuration HMC-wide” on page 239
- “Build a CONFIGxx member” on page 247
- “Process the Display M=CONFIG(xx) command” on page 249
- “Switch IOCDS for next POR” on page 249
- “Specify an IODF for IPL” on page 251

Before the channel subsystem and the operating system can use the configuration that you have defined with HCD, you must build a production IODF from the work IODF.

With the production IODF, you can perform the following tasks in preparation for IPL or for dynamic activation.

- Build an input/output configuration data set (IOCDS) from the production IODF for processors not configured in a z Systems cluster. The configuration can then be used by the channel subsystem.
- Build IOCDSs of central processor complexes (CPCs) configured in a z Systems cluster.
- Manage IPL attributes of central processor complexes (CPCs) configured in a z Systems cluster.
- Build an input data set for the input/output configuration program (IOCP) from the production IODF.
- Create data for input to the JES3 Initialization Stream Checker. This checker program ensures that the data used by MVS is consistent with the data used by JES3. (This task can also be done with a work IODF.)
- Build an OS configuration data set from the production IODF. For VM this is an HCPRIO input data set.
- Verify the configuration described in an IODF against a system.
- Activate the configuration dynamically using the activate function (locally, sysplex wide, or HMC-wide).
- Build a CONFIGxx member for a system from the I/O definitions in an IODF.
- Compare the information in the CONFIGxx member of a system of the sysplex with the existing configuration on that system.

- Switch the IOCDS for the next POR

Build a production IODF

Although HCD validates configuration data as it is entered, a complete validation may not be performed, because data may not be defined at this time. Therefore, a "post-validation" is performed at "Build Production IODF" time. This validation 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.

During the validation HCD invokes the IOCP program to perform checking of the channel packaging rules. Therefore, note that the correct version of the IOCP program must be accessible.

Depending on what is defined in the configuration, the work IODF must contain a definition for at least one operating system, *or* one processor, *or* one switch.

- For an MVS operating system, the IODF must contain at least one EDT and one device.
For a VM operating system, the IODF must contain at least one device as console.
- For a processor, the IODF must contain a definition for at least one channel path, one control unit, and one device. If only receiving CF channel paths are defined for a processor, the control unit and device definitions can be omitted.

Note

A production IODF must have a single extent. If the production IODF has multiple extents, the IPL process results in a WAIT state (wait state code '0B1', reason code '002'). HCD issues error message CBDA009I if a production IODF cannot be built in a single extent.

You can use production IODFs with multiple extents for dynamic activation only. In this case, HCD warns you with message CBDA009I that an IPL with this IODF is not possible, but dynamic activation continues.

To build a production IODF, perform the following steps:

1. On the HCD entry panel, select *Activate or process configuration data*.

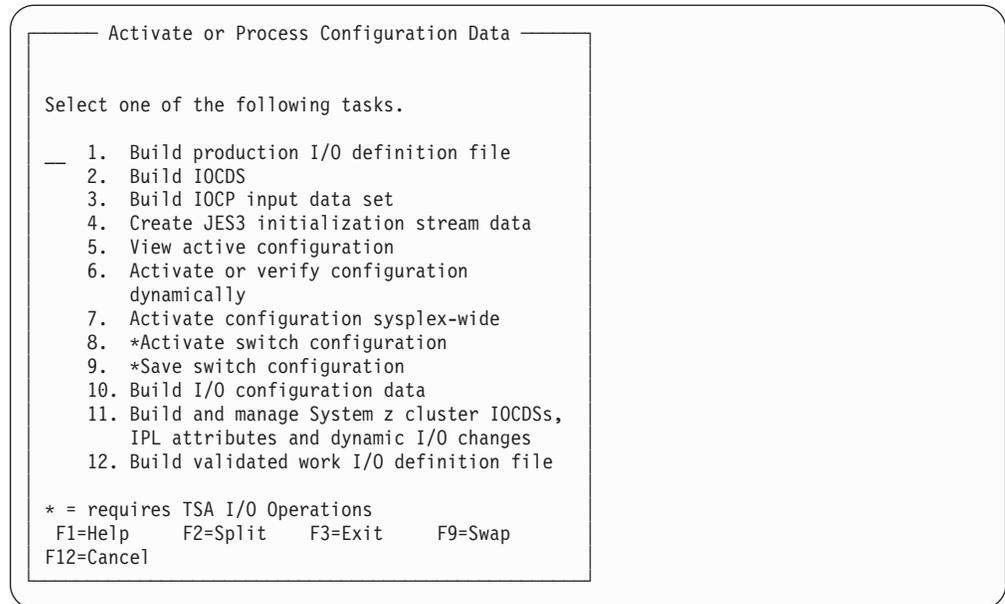


Figure 90. Activate or Process Configuration Data

2. From the resulting panel, select **Build production I/O definition file**. Prior to actually building the production or validated work IODF, HCD updates the work IODF in the following way:
 - For processors, that must be defined for a maximum HSA, it extends the processor configuration to its maximum. This means, HCD ensures that all logical channel subsystems as well as all partitions are defined for the processor, and that each channel subsystem allows for the maximum number of devices per subchannel set.
 - For every primary operating system configuration that specifies the name of a disaster recovery (D/R) site operating system configuration, HCD generates the D/R site OS configuration.

Thereafter, HCD validates the configuration data in the work IODF. If the work IODF is valid, then a production IODF or validated work IODF can successfully be built.

HCD also invokes the IODF checker function that performs a health check of the IODF. In case of defects, HCD issues a severe warning message.

For work IODFs containing XMP processor definitions, before you can build a production IODF, the correct PCHIDs must be defined in the work IODF. You can use the CHPID Mapping Tool to achieve the task to either insert missing PCHIDs or to update PCHIDs in a work IODF. However, inserting or updating PCHIDs into an IODF using the CHPID Mapping Tool is only possible with a so-called **validated work IODF** that you can get in one of two ways:

- a. Use the task **Build validated work I/O definition file**. This task validates a work IODF for correctness and completion, and may issue messages that describe incomplete or erroneous logical definitions. Missing PCHID values are not flagged as errors. If errors occur, correct them and restart this task. As soon as no more errors occur, the output from this task is a validated work IODF.
- b. If you tried to build a production IODF without being aware of one or more missing PCHIDs for XMP processors, but the work IODF satisfies all other validation rules, then the output from **Build production I/O definition file**, too, is a validated work IODF. A message will show all CHPIDs for which the required PCHIDs are missing.

With a validated work IODF, you can use the CHPID Mapping Tool to accomplish the task to update or insert required PCHIDs. Input to this tool is an IOCP input data set. To get this input, now use the task *Build IOCP input data set* from the panel shown in Figure 90 on page 209. This leads you to the *Build IOCP Input Data Set* panel shown in Figure 98 on page 220. Because the input to the CHPID Mapping Tool must be a stand-alone IOCP, in this panel, specify the appropriate option as shown:

Input to Stand-alone IOCP? Yes (Yes or No)

How to proceed using the CHPID Mapping Tool to get PCHIDs inserted or updated in the validated work IODF, see “How to interact with the CHPID Mapping Tool” on page 224. As soon as all PCHIDs are correct in the validated work IODF, the production IODF can be built.

3. If you initially requested activity logging, a panel like the one shown in “Activity logging and change logging” on page 50 is displayed. Enter the activity logging details your installation requires. The Build Production I/O Definition File screen is displayed.

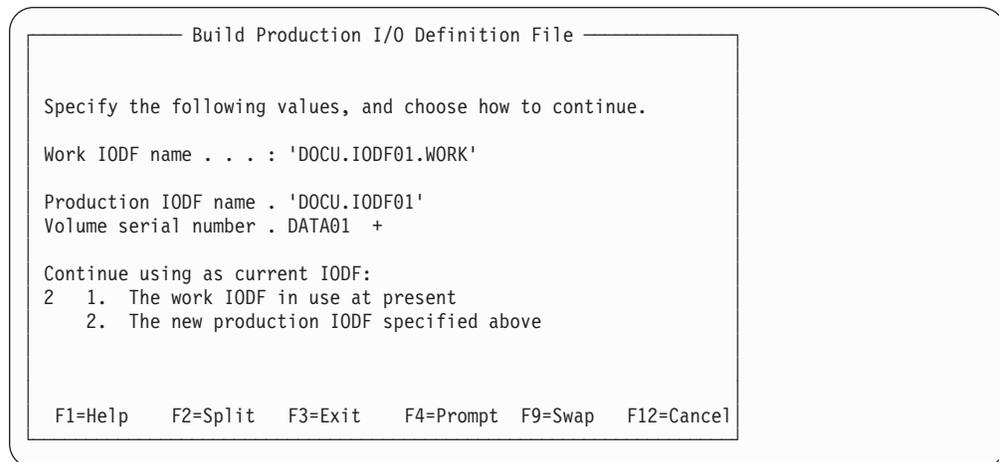


Figure 91. Build Production I/O Definition File

The selection of option Continue using as current IODF controls which IODF is in access after the production IODF has been built. In addition, if you select option 1, The work IODF in use as present, the content of the currently built production IODF is copied to the work IODF. This ensures that the work IODF contains the latest configuration tokens of the IODF, and you can continue to use the work IODF for further updates. If you select option 2, The new production IODF specified above, the content of the production IODF is not mapped into the work IODF. In that case, you should start from the newly built production IODF when performing further changes to the I/O configuration.

4. Specify the name and volume serial number (if applicable) for the production IODF. “IODF naming convention” on page 31 describes the syntax of a production IODF name. If you choose a name without complying to the prescribed syntax of a production IODF name, that IODF can not be used for the IPL and dynamic activate. Moreover, to perform a dynamic activate, the high-level qualifier of the production IODF has to be the same as the one of the IODF used for the previous IPL or dynamic activate.

If the data set name for the production IODF does not adhere to the naming convention for a production IODF, the Confirm Production IODF Name panel is displayed, and you must confirm the IODF name.

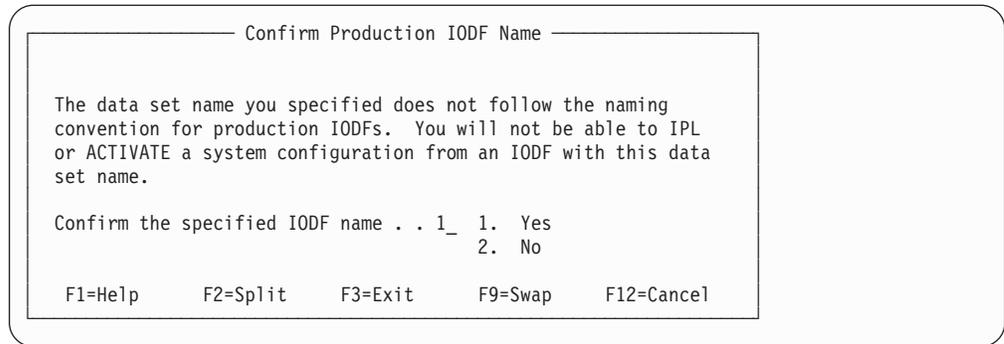


Figure 92. Confirm Production IODF Name

If you use the same name for the new IODF as for an existing IODF, you can replace the existing IODF. In that case, the Confirm Delete I/O Definition panel is shown. Select yes, to confirm deletion of the IODF. Be careful, not to delete the active IODF. If you have specified the name of the active IODF, another confirmation panel is shown that warns you once more about the effect of the chosen name.

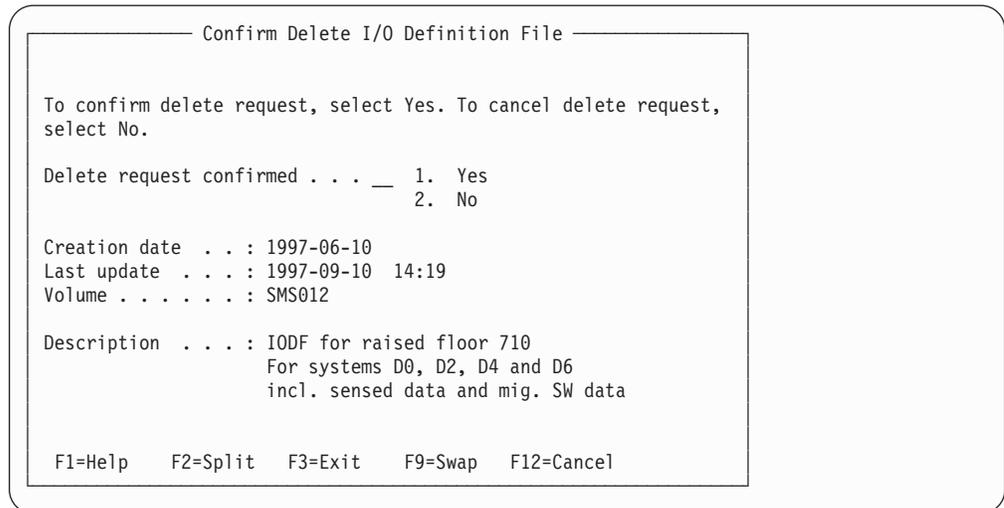


Figure 93. Confirm Delete I/O Definition File

5. After pressing Enter, the Define Descriptor Fields panel appears.

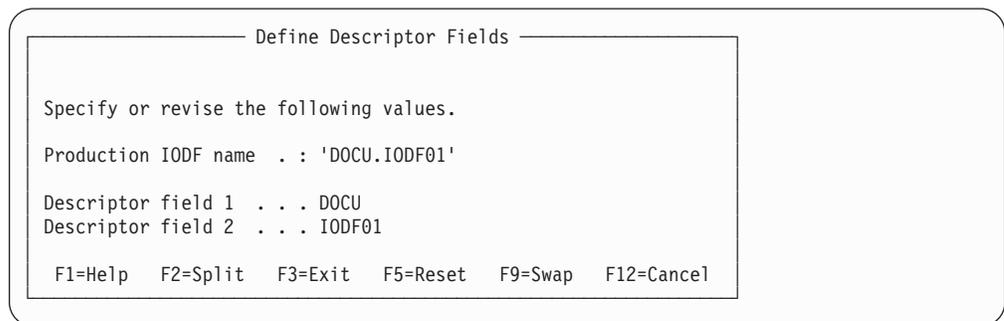


Figure 94. Define Descriptor Fields

Specify the descriptor field 1, 2, or leave the default values. The descriptor fields describe the IODF and will be part of the HSA token. **Attention:** If you specify asterisks (**), equals (==), pluses (++), or minuses (--) for the IODF

suffix in LOADxx, never change the default descriptor field values, because z/OS uses these values to find the current IODF during IPL. Take this relationship also into consideration, if you copy the IODF to a different data set name. For further details refer to *z/OS HCD Planning*. After the production IODF is built, HCD displays a message.

6. If the work IODF has an activity log file defined for the work IODF, it is copied. After the production IODF has been built, HCD informs you that the production IODF has been created.

You can also create a production IODF using the HCD batch facility (for details see “Build a Production IODF” on page 334).

Note:

If the work IODF has an associated MCF, the MCF data set is copied and associated to the production IODF.

Build an IOCDS

When a production IODF has been created, you can build an IOCDS (it can be built only from a production IODF). Processors may have varying numbers of IOCDSs. A particular IOCDS is used at POR time to make the configuration data known to the CSS.

The following procedure is only recommended for processors that do *not* have an SNA address defined, including processors configured in a z Systems cluster. For processors in a z Systems cluster *with* an SNA address defined, use the procedure described under “Build z Systems cluster IOCDSs” on page 215.

While building IOCDSs HCD internally calls the IOCP program. Therefore, note that HCD must be installed in an APF-authorized library.

1. On the *Primary Task Selection* panel, specify the name of a production IODF and select *Activate or process configuration data*.
2. From the resulting panel select *Build IOCDS*. HCD displays the Processor List.
3. On the Processor List, select the processor and press the Enter key. HCD displays the IOCDS List.

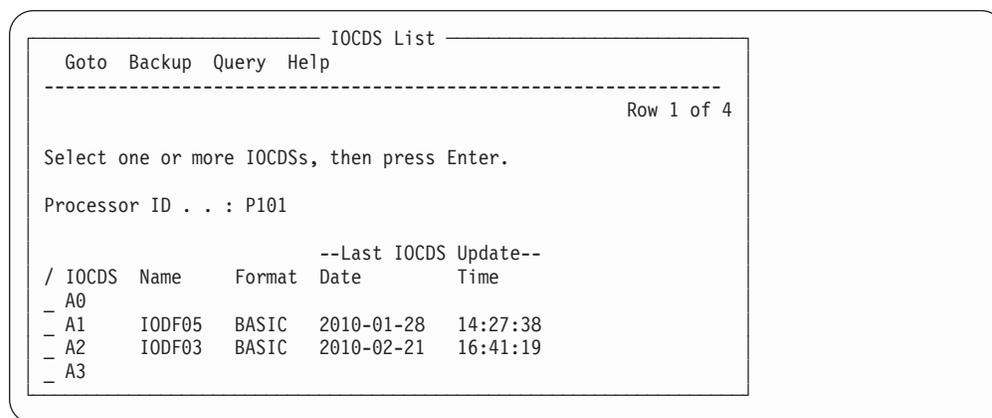


Figure 95. IOCDS List for locally building the IOCDS

The IOCDS list shows those IOCDSs that are built using the currently accessed production IODF.

- Whenever the IOCDS list is invoked, HCD tries to get actual IOCDS data (e.g. date and time of last update) for processors with SNA addresses directly from the support element (SE) and displays it.

IODF data is shown only if the SE does not provide information or where an SNA address is not defined.

A production IODF is updated with data retrieved from the SE if discrepancies between that data and the stored IODF data are detected.

- On the IOCDS List, select the IOCDSs that you want to update and select **Update IOCDS** from the context menu (or action code **u**). HCD displays the Build IOCDS panel.

Build IOCDS

Specify or revise the following values.

IODF name : 'SYS1.IODF02'
 Processor ID : PROC2 IOCDS : A0

Title1 . _____
 Title2 : SYS1.IODF02 - 2009-12-12 14:52

Dualwrite No Remote Write No
 Switch IOCDS for next POR No
 Write IOCDS in preparation of upgrade No

Job statement information
 //WIOCDS JOB (ACCOUNT),'NAME'
 //*
 //*
 //*
 //*
 //*

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

On this panel, you can:

- Into the Title1 field enter identification information you want to be written on the first header line of the IOCP input data set. The first eight characters are used as IOCDS name. This input is used as the MSG1 parameter value of the IOCP ID statement. The batch job passes the MSG1 parameter to the IOCP input data set via the HCDCNTL DD statement (see Figure 145 on page 337 and Figure 146 on page 338).
- Specify the Dualwrite option that describes whether the IOCDS is to be updated on both sides of a physically partitioned processor.
- Specify whether you want to perform a remote or local write of an IOCDS for a processor that has a SNA address defined. The Remote Write option is initialized with 'Yes' if a SNA address is defined to the selected processor. In such a case, HCD initiates a remote IOCDS build and write to the support element with the designated SNA address. If the option is changed to 'No', a local IOCDS build is performed.
- Specify the Switch IOCDS for next POR option, that means whether you want to make this IOCDS the active one for the next power-on reset (POR).
- Specify whether to Write IOCDS in preparation of upgrade. This specifies whether an IOCDS is to be written regardless of processor type. This is useful to prepare for a processor upgrade.

If Yes is specified, an IOCDS for the selected processor is written regardless of the processor type. For a list of processors that support writing an IOCDS,

in preparation for a processor upgrade, or for which such an IOCDS can be written, see “Supported hardware report” on page 425.

Note: If, as a result of a processor upgrade, an IOCDS download is not possible you can, after having built the production IODF, create an IOCP input data set that can be used with the stand-alone IOCP to generate an IOCDS for use with POR.

- Change the job statement information to meet the installation needs. Note that a batch job to build an IOCDS must run on the processor on which the IOCDS is to be updated. (See “Job statement information used in panels” on page 72 for a description of the job control information that you need to specify when you build an IOCDS.)

In a multiprocessor JES environment, be sure to specify the JES command and/or job class to ensure that the job runs on the correct processor(s).

The recommended region size is 2 MBytes more than the IOCP needs. For the region size required by IOCP, refer to the *IOCP User's Guide* for your processor.

6. When an IOCDS is built, a record is written for the processor configuration. If you build a new IODF from an existing IODF, the records are copied to the new IODF. When you build IOCDSs from this new IODF, the IOCDSs from the old IODF are also shown on the IOCDS list. Because the batch job requires exclusive use of the production IODF for processors that have *no* SNA address specified, you have to either leave the HCD session or change the currently accessed IODF to run the submitted job.

You can also invoke the *Build IOCDS* task in batch mode, see “Build an IOCDS or an IOCP input data set” on page 335.

Note:

1. Unlike writing an IOCDS using the IOCP program, the HCD process generates no IOCP report when using the *Build IOCDS* function.
2. When a processor has been upgraded in the IODF, the old IOCDS status data in the IODF is deleted.

Using this procedure for processors with an SNA address:

If you use this procedure for processors in a z Systems cluster *with* an SNA address defined, HCD does the following:

- HCD writes the IOCDS for that processor to the support element with the designated SNA address (remote IOCDS build).
- If you run the job under MVS/ESA SP Version 5, OS/390, or z/OS, HCD writes the IOCDS for that processor to the support element with the designated SNA address.

To run the batch job, you do not have to leave the HCD session. If the processor has defined an SNA address, HCD assumes that it is part of a z Systems cluster. In this case, the job can immediately start without the need to free the currently accessed IODF.

HCD tries to update the IOCDS record but is not able to because the IODF is still allocated by the HCD dialog. This results in an error message on the console log. To avoid this error message, HCD offers the profile option of bypassing the IODF information update (see “Bypass IODF information update for SNA processor” on page 26).

Build z Systems cluster IOCDs

The following procedure describes how to build an IOCDs for processors in a z Systems cluster *with* an SNA address defined.

To build IOCDs within a z Systems cluster:

- The SNA address has to be defined for a CPC configured in a z Systems cluster
- Specific RACF authority has to be attained (for details on required access authority, refer to “Security-related considerations” on page 359).
- The operating system must not be running as a guest under z/VM.

Perform the following steps:

1. On the **Primary Task Selection** panel, select **Activate or process configuration data** and from the resulting panel select **Build and manage z Systems cluster IOCDs, IPL attributes and dynamic I/O changes**. The **z Systems Cluster List** is displayed:

```
Goto Query Help
-----
z Systems Cluster List                               Row 1 of 18
Command ==> _____ Scroll ==> PAGE

Select one or more CPCs, then press Enter.

-----CPC----- IODF
/ SNA Address      Type  Model Processor ID
[S] IBM390PS.DAN2   2094 S28   DAN2
[S] IBM390PS.ECL2   2097 E40   ECL2
- IBM390PS.G14     2084 B16   G14
- IBM390PS.G15     2084 D32   G15
- IBM390PS.H05     2097 E26   H05
- IBM390PS.H37     2097 E26   H37
- IBM390PS.H42     2097 E26   H42
- IBM390PS.P0L1    2096 S07   P0L1
# IBM390PS.P0ZGMR04
- IBM390PS.P0000H27 2097 E56   H27
# IBM390PS.RAP     2066 004
- IBM390PS.R17     2817 M32   R17
[V] IBM390PS.R35    2817 M49   R35
- IBM390PS.R37     2817 M32   R37
```

Figure 96. z Systems Cluster List

This panel shows all CPCs configured in a z Systems cluster. They are identified by the SNA address of their support element and displayed together with their Type and Model as well as the Processor ID in the IODF. The SNA address has been specified in the processor definition task for the IODF processor definitions and enables the relation to the configured CPCs.

A disabled sign (#) in the action entry field can be due to:

- SNA address not defined in the IODF. In this case, the IODF Processor ID shows no value. Either define the SNA address for a corresponding processor in the accessed IODF or use another IODF.
- SE (support element) of CPC did not respond. In this case, the CPC Type and Model fields show no values.

A processor ID followed by two dots (..) indicates that this SNA address has been defined for several processors in the IODF. The first processor ID (in alphabetical order) with the SNA address is displayed. If you want to apply any of the group actions on another processor, use **Select other processor configuration** from the context menu (or action code *p*).

- On the *z Systems Cluster List*, select the CPCs for which you want to build and manage the IOCDSs and *Work with IOCDSs* from the context menu (or action code **S**). HCD displays the IOCDS List (shown with sample data):

```

Goto Query Help
-----
IOCDS List                               Row 1 of 8 More:  >
Command ==> _____ Scroll ==> PAGE

Select one or a group of IOCDSs, then press Enter.

-----Token Match----- Write
/ IOCDS   Name     Type   Status  IOCDS/HSA IOCDS/Proc. Protect
- A0.DAN2  052DAN2  LPAR  POR     Yes       Yes       Yes-POR
- A0.ECL2  062AECL2 LPAR  Alternate No        No        No
A A1.DAN2  021ADAN2 LPAR  Alternate No        No        No
- A1.ECL2  054AECL2 LPAR  Alternate No        Yes       No
- A2.DAN2  025ADAN2 LPAR  Alternate No        No        No
- A2.ECL2  059AECL2 LPAR  Alternate No        No        No
- A3.DAN2  026ADAN2 LPAR  Alternate No        No        No
- A3.ECL2  062AECL2 LPAR  POR     No        No        Yes-POR

```

Figure 97. IOCDS List invoked from the *z Systems Cluster List*

On the IOCDS List, all applicable IOCDSs of the selected CPCs are displayed and arranged in ascending order by IOCDS names (starting, for example, with A0-IOCDSs, A1-IOCDSs). This list enables you to apply the IOCDS functions as group actions against one or several IOCDSs for all selected processors.

The data displayed is retrieved directly from the support elements. If, however, the support element does not answer, HCD displays the data saved in the IODF and issues a message accordingly.

A production IODF will be updated with data retrieved from the support element if discrepancies between that data and the stored IODF data are detected. A work IODF will remain unchanged.

The Type field contains one of the following types of power-on reset modes to be used with the I/O configuration defined in the IOCDS: S/370, ESA/390, or LPAR.

The Status field indicates the status of the IOCDS:

Alternate

not to be used at the next POR

POR

to be used at the next POR

Invalid

IOCDS is opened for update

The Token Match-IOCDS/HSA field indicates whether the IOCDS token matches the current HSA token. If Yes is shown, it means that the IOCDS has been built by HCD, and that it matches the current I/O configuration - either because this IOCDS was used for the last POR, or the matching configuration has been activated dynamically.

The Token Match-IOCDS/Proc. field indicates whether the IOCDS token matches the processor token in the IODF, currently used in the HCD dialog. If Yes is shown, the IOCDS has been built from the IODF currently used in the HCD dialog.

- On the IOCDS List, you can select the following actions from the context menu:
 - Use the **Update IOCDS** action (or action code **U**) to build or update the selected IOCDSs with the I/O configuration data from the currently accessed production IODF. See step 4 on page 217 on how to proceed.

- Use the *Switch IOCDS* action (or action code *s*) to mark an IOCDS as the IOCDS that is used for the next POR. The Status field will be set accordingly.

You can only switch to an IOCDS that has an IOCDS/HSA token match or to an IOCDS of a processor that is not activated ('POR-required' status).

- Use the *Enable write protection* or *Disable write protection* action (or action codes *e* and *w*) to allow or prohibit updating the selected IOCDSs of the designated CPCs. The Write Protect field will be set accordingly.

Use the F20=Right key to move the screen to the right to see information such as date and time of the last IOCDS update and the IOCDS configuration token.

4. If you select the *Update IOCDS* action, HCD displays the Build IOCDSs dialog.

```

Build IOCDSs
-----
Row 1 of 1
Command ==> _____ Scroll ==> PAGE
Specify or revise the following values.
IODF name . . . . . : 'SYS1.IODF01'
Title1 . _____
Title2 : SYS1.IODF01 - 2011-02-24 09:40
IOCDS      Switch IOCDS  Write IOCDS in
A1.DAN2    No           preparation of upgrade
***** Bottom of data *****
F1=Help    F2=Split    F3=Exit    F5=Reset   F7=Backward
F8=Forward F9=Swap     F12=Cancel F22=Command

```

On this panel you can:

- Enter identification information you want to be written on the first header line of the IOCP input data set in the Title1 field.
- Specify the Switch IOCDS option, if you want to make this IOCDS the active one for the next power-on reset (POR).
- Specify whether to Write IOCDS in preparation of upgrade. This specifies whether an IOCDS is to be written regardless of processor type. This is useful to prepare for a processor upgrade.

If Yes is specified, an IOCDS for the selected processor is written regardless of the processor type. For a list of processors that support writing an IOCDS, in preparation for a processor upgrade, or for which such an IOCDS can be written, see “Supported hardware report” on page 425.

Note: If, as a result of a processor upgrade, an IOCDS download is not possible you can, after having built the production IODF, create an IOCP input data set that can be used with the stand-alone IOCP to generate an IOCDS for use with POR.

5. After pressing the Enter key on the Build IOCDSs panel, the Job Statement Information panel is displayed. Specify the information for the batch job that HCD generates to build the IOCDSs.

The recommended region size is 2 MBytes more than the IOCP needs. For the region size required by IOCP, refer to the *IOCP User’s Guide* for your processor.

Manage z Systems cluster IPL attributes

For IPL operations for CPCs configured in a z Systems cluster, you can:

- Display the IPLADDR and IPLPARM attribute values of the last and for the next IPL.
- Modify IPLADDR and IPLPARM attribute values to be used for next IPL.

Perform the following steps:

1. On the *Primary Task Selection* panel, select *Activate or process configuration data* and from the resulting panel, select *Build and manage z Systems cluster IOCDs, IPL attributes and dynamic I/O changes*. The *z Systems Cluster List* is displayed (see Figure 96 on page 215).
2. On the *z Systems Cluster List* select the CPCs for which you want to view and modify IPL attributes.
3. Select the *Work with IPL attributes* action from the context menu (or action code **i**). HCD displays the IPL Attribute List.

```

----- IPL Attribute List -----
                                IPL Attribute List      Row 1 of 6 More:  >
Command ==> _____ Scroll ==> PAGE

Update the values to be used for the next IPL and press Enter. To view the
values used for the last IPL, scroll to the right.

Processor  Partition  IPL      -----Next IPLPARM-----
ID         Name       ADDR    IODF  LOADxx  Prompt/Msg  Nucl  Unformatted
          Name       ADDR    Dev   Suffix  Option     Suffix IPLPARM

H05       LIN1       4800    1     --     -         -     1
H05       LIN2       4801    --    --     -         -     --
H05       BVM1       A140    0EAO  --     -         -     0EAO
H05       COM1       1A00    1A10  F1     -         -     1A10F1
H05       COM2       1A01    1A10  FA     A         2     1A10FAA2
H05       AVM4       A100    CONS  SY     S         G     CONSSYSG

***** BOTTOM OF DATA *****

```

The IPL Attribute List displays the IPLADDR and IPLPARM attribute values for all selected processor definitions and their partitions that are obtained from the support element of the associated CPCs.

4. On the *IPL Attribute List*, view or modify the attribute values for IPLADDR and IPLPARM.

Use F20=Right to move the screen to the right to view the IPL attributes used for the last IPL.

The *IPL ADDR* column shows the LOAD address used for next IPL.

The *Next IPLPARM* column shows the LOAD parameter used for z/OS and is a concatenation of the following attributes: IODF Device, LOADxx Suffix, Prompt/Message Option, and Nucleus Suffix. An additional column shows unformatted IPL parameters. You can use these unformatted values to specify non z/OS IPL parameters, for example, for z/VM or Linux on System z for the next IPL. This unformatted string can be up to eight characters long. HCD does not perform any semantic checks of these values, as the content depends on the operating system for which they are used.

When providing input to the unformatted IPL parameters, you must specify the value at the correct position in the string. For non-specified leading or intermediate characters, use a period (.) instead of a blank, since blanks are removed by ISPF.

You cannot type values into both the left four columns and the right *Unformatted* column of *Next IPLPARM*. If you provide values into the *Unformatted* column, they are transferred into the left four columns.

The *IPL ADDR* and/or the *Next IPLPARM* values for the next IPL are taken, when 'Use dynamically changed IPL address' and/or 'Use dynamically changed IPL parameter' are selected on a LOAD profile that is used to initiate an IPL, or to activate an operating system. This enables you to change the values of IPLADDR and IPLPARM without updating the profile.

Build an IOCP input data set

Sometimes it is necessary to build an IOCP input data set:

- If you have to use the stand-alone IOCP program, which does not support direct access to the production IODF.
- If you want to create a backup on tape in case you need to recover the contents of the IOCDS in the service processor.
- If you need an IOCP input data set as input to the CHPID Mapping Tool in order to map CHPIDs to PCHIDs for XMP processors.
- If you need an IOCP input data set for a processor on which there is no HCD running (e.g. a new processor). Also, if you upgrade a processor to a model that results in a new IOCP SYSTEM value and the processor does not support an IOCDS write in preparation for a processor upgrade, you have to run the stand-alone IOCP program.

How to build an IOCP data set

1. On the *Primary Task Selection* panel, specify the name of a production IODF and select *Activate or process configuration data*.
2. On the resulting panel select *Build IOCP input data set*. HCD displays the Available Processors screen.
3. On the Available Processors screen, select the processor for which you want to build the IOCP input data set. HCD displays the Build IOCP Input Data Set dialog.

```

Build IOCP Input Data Set
-----
Specify or revise the following values.

IODF name . . . . . : 'BOKA.IODF08'
Processor ID . . . . . : CF01
Title1 : _____
Title2 : BOKA.IODF08 - 2009-07-02 15:42

IOCP input data set
-----
Input to Stand-alone IOCP? Yes (Yes or No)

Job statement information
//WIOCP JOB (5765), 'BOKA', NOTIFY=BOKA, CLASS=A, MSGCLASS=X, REGION=5M
//JOB LIB DD DSN=HCDTEST.IZPIOCP.FALC, DISP=SHR
//GO.HCDPROF DD DSN=HCDTEST.PROFILE(MIGENH51), DISP=SHR
//*

```

Figure 98. Build IOCP Input Data Set

4. On this panel you can:

- Enter the identification information you want to be written on the first header line of the IOCP input data set in the Title1 field.
- Specify the name of the IOCP input data set. The IOCP input data set will automatically be allocated (record length 80, record format fixed block). If the data set already exists, you will be asked to confirm replacing it with the new one.
- Specify whether to build the IOCP data set for stand-alone IOCP.

Yes

This is the default. The generated IOCP statements can be used as input to the stand-alone IOCP program or to the CHPID Mapping Tool.

Note: You may not be able to use such a generated IOCP input data set for the migration function of HCD because, for example, the unit name of control units and device types can be truncated due to IOCP restrictions.

No

The IOCP input data set is built using the IOCP changes described in “IOCP enhancements” on page 221 and generating the extended migration parameters and statement (if the profile statement MIGRATE_EXTENDED is set to YES) as described in “IOCP input data sets using extended migration” on page 221. Note that if you try to process these IOCP statements with the stand-alone IOCP program, you may run into problems, because the program may not accept the generated syntax.

- Change the job statement information to meet the installation needs. With JCL overwrite statements you can modify the EXEC procedure that is invoked. You can, for example, specify the HCD profile using the job step name GO. (See “Job statement information used in panels” on page 72 for a description of the job control information that you need to specify when you build an IOCP input data set.) Ensure that the batch job runs in a region with at least 4 MBytes.

TOK=value

Configuration programs use this keyword to forward information to the CPC which is required to enable the dynamic I/O configuration capability of any

resulting IOCDs. This keyword is not intended for direct user input. The contents needs not relate to the target processor. It just must match the token in the IOCDs/HSA and the currently active IODF.

You can also invoke this task in batch mode. See “Build an IOCDs or an IOCP input data set” on page 335.

Note:

You should never change an IOCP input file generated by HCD and use it to write an IOCDs. If changes are necessary, use HCD to regenerate the IOCP input.

IOCP enhancements

The generated IOCP data set contains control unit and device types of 8 characters and a device model of up to 4 characters. Such an IOCP input data set can be processed by IOCP (with APAR OW13343) and remigrated to HCD without the need to correct the control unit and device types that exceed the 5 character UNIT and 2 character MODEL value limitation.

It also now contains an all-character readable token which allows the user to preserve the dynamic capability when performing a stand-alone IOCP run on a z Systems cluster CPC using IOCP input from diskette.

Important Note:

It may not be possible to remigrate an IOCP input data set generated by HCD back into the IODF. The reasons are:

- HCD uses the High Level Assembler program for parsing the IOCP statements. The High Level Assembler earlier than V1.5 is restricted to 255 characters for any keyword value. IOCP statements, however, may contain keywords with a value greater than 255 characters. High Level Assembler V1.5 removes this restriction.
- HCD keeps additional data for a processor configuration that is not contained in an IOCP input data set. This data may be used for validation and, therefore, missing at the migrate step leading to validation errors. For example, the partition usage is defaulted to CF/OS. For a shared CF peer channel, this may lead to a validation error, because only a CF partition may be specified in the access or candidate list.
- Since the IOCP data are only a subset of the processor configuration data, you may lose this additional configuration data if you update a processor configuration from an IOCP input data set.
- IOCP data sets do not contain devices connected to a processor with a null device candidate list because of IOCP rules. If the device is connected to another processor with the same control unit, this is an ambiguous configuration and is not migrated.

For updating the IODF via I/O configuration statements, it is recommended to use the extended I/O configuration statements of HCD instead of an IOCP input data set (see “IOCP input data sets using extended migration”).

IOCP input data sets using extended migration

As described in “Updating parts of a configuration by migrating input data sets” on page 313, HCD introduces an extended migration to allow you to define your complete configuration without using the ISPF front end dialog.

For example, the extended migration allows you to define a switch with its ports or define serial numbers and descriptions for devices and control units by editing your input data sets and migrating them into HCD.

Analogously, when building an IOCP input data set from an IODE, information is generated that describes the additional parameters of the configuration objects (if the prerequisites under “Prerequisites to exploit the extended migration” are met). Within the generated IOCP input data set, the additional parameters and control statements are shown as comments with special HCD tags so that they can be processed by the IOCP program. When re-migrating such an IOCP input data set to HCD, the tagged comments are identified by HCD and migrated correspondingly.

If you want to use the input data set for both, IOCP processing and HCD migration, the new records must apply to the following rules, so that they can be processed by both programs:

- The new parameters start with the string `*$HCDC$` in column 1.
- The new SWITCH statement starts with the string `*HCD` in column 1.
- The IOCP statement does not have any comment.
- The additional HCD tagged records follow immediately the last record of the corresponding IOCP statement.
- The first keyword starts at column 16.
- The last operand is not followed by a comma.
- There is no comment to the right of the operand.

Prerequisites to exploit the extended migration

To generate the additional keywords during IOCP data set build, note the following prerequisites:

- Specify the following entry in the HCD profile:
`MIGRATE_EXTENDED = YES`

When you specify `MIGRATE_EXTENDED = NO` (which is default), the additional keywords are not generated during IOCP build. In addition, when remigrating the IOCP input data sets, the migration function ignores the commented `'*$HCDC$'` and `'*HCD'` tags.

The HCD profile is explained in “Defining an HCD profile” on page 19.

- When building IOCP input data sets, you have to set the option Input to Stand-alone IOCP to No on the Build IOCP Input Data Set panel. See “Build an IOCP input data set” on page 219 for a description of the new option.

Example of an IOCP input data set

Figure 99 on page 223 shows you an example of a generated IOCP input data set with the new parameters. Note that each new parameter starts with an `*$HCDC$` in column 1. The new switch control statement starts with `*HCD` in column 1.

```

ID      MSG1='IOCDNAM',MSG2='BOKA.IODF03 - 95-07-21 16:00',      *
      TOK=('TWO',0000000190009672160057040095202F00000000,0000*
      0000,'95-07-21','16:00:57','BOKA','IODF03')
*$HCDC$      DESC='Cluster(099) test floor'
*$HCDC$      SERIAL='1044009672'
*$HCDC$      SNAADDR=(USIBMSC,TWO)
RESOURCE PARTITION=((CF001,3),(MVSSMAL,2),(PRIME,1))
*$HCDC$      DESCL=('Coupling facility','MVS 5.2.0 System1','Producti*
      on CF image')
CHIPID PATH=(10),PARTITION=((CF001),(CF001)),TYPE=CFR
*$HCDC$      DESC='Receiver'
CHIPID PATH=(13),PARTITION=((PRIME),(PRIME)),TYPE=CFS
*$HCDC$      TPATH=(TWO,10,FFFE,FFFE)
*$HCDC$      DESC='Sender'
CHIPID PATH=(20),PARTITION=((MVSSMAL),(MVSSMAL)),SWITCH=AB,      *
      TYPE=CNC
*$HCDC$      SWPORT=((AA,C0))
*$HCDC$      DESC='Channel for DASD'
CHIPID PATH=(21),PARTITION=((MVSSMAL),(MVSSMAL)),TYPE=CNC
*$HCDC$      DESC='Channel for DASD'
CHIPID PATH=(25),PARTITION=((PRIME),(PRIME)),TYPE=CNC
*$HCDC$      SWPORT=((AB,80))
*$HCDC$      DESC='Switch connection'
CHIPID PATH=(26),PARTITION=((MVSSMAL),(MVSSMAL)),TYPE=CNC
*$HCDC$      SWPORT=((AA,C5))
*$HCDC$      DESC='Switch connection'
CNTLUNIT CUNUMBR=0005,PATH=(25),UNITADD=((00,001)),UNIT=9032-3
*$HCDC$      SWPORT=((AB,FE))
*$HCDC$      SERIAL='1021-CU511'
*$HCDC$      DESC='SWITCH AB'
CNTLUNIT CUNUMBR=0006,PATH=(26),UNITADD=((00,001)),UNIT=9033
*$HCDC$      SWPORT=((AA,FE))
*$HCDC$      SERIAL='1021-CU510'
*$HCDC$      DESC='SWITCH AA'
CNTLUNIT CUNUMBR=000F,PATH=(20,21),UNITADD=((08,008)),      *
      LINK=(A1,A2),CUADD=3,UNIT=3995-151
*$HCDC$      SWPORT=((AA,C3),(AB,82))
*$HCDC$      SERIAL='5512003330'
*$HCDC$      DESC='DASD on Q4-B3'
CNTLUNIT CUNUMBR=FFFE,PATH=(13),UNIT=CFS
IODEVICE ADDRESS=(080,004),MODEL=151,UNITADD=08,      *
      CUNUMBR=(000F),STADET=Y,UNIT=3995
IODEVICE ADDRESS=110,MODEL=3,UNITADD=00,CUNUMBR=(0005),      *
      STADET=Y,UNIT=9032
*$HCDC$      SERIAL='1021-CU511'
*$HCDC$      DESC='SWITCH AB'
IODEVICE ADDRESS=120,UNITADD=00,CUNUMBR=(0006),STADET=Y,      *
      UNIT=9033
*$HCDC$      SERIAL='1021-CU510'
*$HCDC$      DESC='SWITCH AA'
IODEVICE ADDRESS=(FFFE,002),CUNUMBR=(FFFE),UNIT=CFS
*$HCDC$      SWITCH SWID=AA,SERIAL='1021-CU510',DESC='SWITCH AA',      *
      PORT=((C0,CE),(FE,FE)),SWPORT=((CA,AB,C6)),UNIT=9033
*$HCDC$      SWITCH SWID=AB,SERIAL='1021-CU511',DESC='SWITCH AB',      *
      PORT=((80,F0),(FE,FE)),SWPORT=((C6,AA,CA)),MODEL=3,      *
      UNIT=9032

```

Figure 99. Example of an input data set for migration enhancements

Using the IOCP data set as input for the CHPID Mapping Tool

HCD will allow generating an IOCP deck that does not contain any or all necessary PCHID values. You can use this IOCP deck as input to the CHPID Mapping Tool in order to have PCHIDs inserted or updated. The CHPID Mapping

Tool then generates a new IOCP input deck containing the assigned PCHID values. You can then migrate the updated PCHIDs into a validated work IODF (see also “How to interact with the CHPID Mapping Tool”).

How to interact with the CHPID Mapping Tool

Correct PCHIDs are required in the configuration for XMP processors before you can successfully build a production IODF. The task of adding or updating required PCHID information for a work IODF for XMP processors is eased by an interaction between HCD and the CHPID Mapping Tool (CMT). Prerequisite for this task is a so-called *validated work IODF* that you can get in one of two ways described in “Build a production IODF” on page 208.

You can download the CHPID Mapping Tool from the Internet. It runs on a workstation.

Process overview

Input to the CMT is the hardware configuration file (CFReport¹) of your machine and a valid IOCP input file (with missing or obsolete PCHIDs).

Output from the CMT is again an IOCP input file that now has all missing or updated PCHID values filled in. Upload this IOCP input file and re-import it into the validated work IODF using the HCD primary task *Migrate configuration data*.

Via this migration task, it is possible to update a validated work IODF with the PCHID values that have been written by the CMT into the IOCP input file. Other changes on the validated work IODF are not possible without losing the status of a validated work IODF. A PCHID migration is only possible to a validated work IODF. Since PCHID migration changes the IODF, the IODF status is reset to 'not validated'. All functions that allow read-access to a work IODF are also possible for a validated work IODF. Activate functions are not possible, except for building an IOCP deck that can be used as input to the CMT. Only after all PCHIDs have been inserted into the validated work IODF, you can successfully build a production IODF.

How to insert or update PCHIDs

Here are the detailed steps you need to perform to insert or update PCHIDs in a validated work IODF.

1. Create a validated work IODF with one of the two methods described in “Build a production IODF” on page 208. Your validated work IODF may lack at least one PCHID that you need to insert or may contain obsolete PCHIDs that you want to update.
2. Go back to the *Activate or Process Configuration Data* menu shown in Figure 90 on page 209 and use task *Build IOCP input data set* to export the I/O configuration from the validated work IODF to an IOCP data set (with PCHIDs still missing or obsolete). The hardware configuration token is passed with the IOCP statements (TOK keyword). This token is used to assure that during the process of assigning PCHID values the contents of the IODF is not changed.

Download this IOCP data set to the workstation where the CMT is running.

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.

3. Use the CHPID Mapping Tool with the downloaded IOCP data set. For information on how to use the CHPID Mapping Tool, refer to the online help and the related documentation.

The output of a successful CMT run is again an IOCP data set which contains the original I/O definitions together with inserted and/or updated PCHID values. The original hardware configuration token is still contained in the generated statements.

4. Upload the new IOCP data set to the host and use the HCD primary task *Migrate configuration data* to import the PCHIDs from the updated IOCP data set into the validated work IODF. During this task, you select

migrate option ---> 3. PCHIDs

from Figure 135 on page 287 in “Migrating input data sets using the HCD dialog” on page 287.

When importing these statements into the validated work IODF via the migration process for PCHID migration, HCD verifies that the token passed with the IOCP statements match the token stored in the IODF. If this is the case, and if the logical I/O definition described by the imported IOCP statements does not differ from the IODF data, HCD writes the PCHID values into the IODF. If the token does not match, for example, because the IODF has been updated in the meantime, a PCHID migration will not be performed. In this case you must start the process from the beginning.

Note: The IOCP input data set may contain keyword values which exceed the 255 character limitation of the assembler program used by HCD for parsing the IOCP statements. This may be the case for the PARTITION (PART) keywords on the RESOURCE statement and on the CHPID statements for spanned channel paths. The affected I/O configuration statements must be deleted in order to perform a successful PCHID migration. However, note that the High Level Assembler V1.5 removes the 255 character limitation.

5. If a PCHID migration has been successfully done, you can invoke the *Build Production IODF* task (again). HCD now builds a production IODF that contains all the data that is required to write the I/O configuration data set (IOCDs) via the IOCP program to the Support Element (SE) of the machine, ready to be used for the next IML.

Note: 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 116. Thus you can avoid to let the CHPID Mapping Tool run into an error, because it cannot find the PCHID.

To support the algorithm of mapping the logical CHPID definitions to physical channels, a CMT user can specify priorities to the control units (CU priorities). It is possible to preserve these values across different invocations of the CMT. For this purpose, the CMT passes the CU priorities as special comments to HCD. HCD takes these comment lines and stores them in the IODF attached to the corresponding processor. When generating an IOCP input file for the CMT, HCD includes these comments into the generated statements. HCD does not make these comments visible in the HCD dialog or in the reports.

Create JES3 initialization stream checker data

Because JES3 does not access the IODF directly, it has to be checked whether JES3 I/O and MVS I/O definitions are the same. It is essential that these definitions are consistent. Each time you run this task, the JES3 initialization stream checker data is stored in a data set, thus allowing JES3 to check the previous mentioned definitions and to detect inconsistencies among them.

You can start the task from a work IODF as well as from a production IODF. However, it is recommended to use a production IODF to ensure that the same information is used for IPL.

1. On the *Primary Task Selection* panel, select *Activate or process configuration data*.
2. On the resulting panel select *Create JES3 initialization stream data*. HCD displays the Create JES3 INISH Stream Checker Data panel.

Create JES3 INISH Stream Checker Data

Specify or revise the following values.

JES3 initialization stream checker data set name

MVS configuration ID OPSYS01 +
EDT identifier A1 +

3. Specify the required values.
The output data set will automatically be allocated (record length 80, record format fixed block). Depending on whether you specify the data set name as sequential or partitioned, the data set will be either sequential or partitioned. It is recommended to specify a partitioned data set (PDS), because this is required by the JES3 initialization stream checker.
If the data set already exists, you will be asked to confirm replacing it with the new one.

You can also invoke this task in batch mode. See “Build I/O configuration data” on page 339 for a description of the job control information that you need to specify when you build JES3 initialization stream checker data.

Build I/O configuration data

You can use HCD to create an I/O configuration data set containing either:

- an OS configuration
 - a processor configuration
 - a switch configuration
 - FCP device data.
1. On the *Primary Task Selection* panel, select *Activate or process configuration data*.
 2. From the resulting panel select *Build I/O configuration data*. HCD displays the *Build I/O Configuration Data* dialog.

```

Build I/O Configuration Data

Specify or revise the following values.

IODF name . . . . . : 'DOCU.IODFA0.WORK'

Configuration type . . 2  1. Processor
                        2. Operating System
                        3. Switch
                        4. FCP Device Data

Configuration ID . . . _____ +
Output data set . . . _____

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

```

3. Specify one of the available configuration types:

- **Options 1 - 3:**

Specify the identifier of the configuration to be used and a name for an output data set to contain the configuration statements according to the specified configuration type. The data set is automatically allocated (record length 80, record format fixed block).

Specifying an asterisk (*) in the **Configuration ID** field generates all configurations of the specific type to the output I/O configuration data set. If the data set already exists, you will be asked to confirm replacing it with the new one.

For **Option 3**, HCD offers an additional feature:

If you specifying an asterisk (*) in the **Configuration ID** field for a switch configuration, and with profile keyword SHOW_CONFIG_ALL set to YES, HCD additionally generates configuration statements for control units and devices without a processor and OS connection.

- **Option 4:**

HCD exports the FCP device configurations for a specific processor from the currently accessed IODF as comma-separated values into a CSV output format which you can use as input to the WWPAN Prediction Tool to assign world-wide port names to virtualized FCP ports. This HCD output file is also referred to as *FCP SAN configuration template file*.

Specify the desired processor ID in the **Configuration ID** field. The data set will be automatically allocated (record length 132, record format fixed block). If the data set already exists, you will be asked to confirm replacing it with the new one.

You can also invoke this task in batch mode. See “Build I/O configuration data” on page 339 for a description of the job control information that you need to specify.

Verify an I/O configuration

HCD allows you to check the definitions in your IODF against the actual configuration as sensed from the active system. This can be done using TSA I/O Operations (see “Prerequisites” on page 9 for the prerequisites) or zDAC.

The verify function results in a list of all sensed paths in comparison to the defined paths. Using a filter, this usually extensive list can be reduced to the data of interest. The list can be saved and/or printed. The verification can also be carried out as a batch job resulting in an I/O Path Report (see the example Figure 189 on page 424 in section “Print configuration reports” on page 343).

Verifying a configuration against the local system

1. To verify the I/O configuration of the local system select *Activate or process configuration data* from the primary task level. From the resulting panel select *Activate or verify configuration dynamically*. The Activate or Verify Configuration panel is displayed.

Note: For the verify function on the Activate or Verify Configuration panel to be available, the processor configuration from which the active IOCDS was built must match the configuration in the IODF used for IPL (token match).

2. Select the *Verify active configuration against system* task to compare the system against the active IODF. Select the *Verify target configuration against system* task to compare the system against the accessed IODF. The Identify System I/O Configuration panel is displayed.

```

Identify System I/O Configuration

Specify or revise the following values. Press ENTER to
continue.

IODF to be used . . . . : IODFST.IODF4F
Processor ID . . . . . FR38LPAR +
Partition name . . . . F38H +
OS configuration ID . B710 +

I/O Cluster name . . . . _____ + (only for Build CONFIGxx)

```

Figure 100. Identify System I/O Configuration

3. Specify the processor ID and OS configuration ID. If the specified processor is in LPAR mode, you must also specify a partition name.

Verifying a configuration against a system in the sysplex

1. Select *Activate or process configuration data* from the primary task level. From the resulting panel select *Activate configuration sysplex-wide*. The *Active Sysplex Member List* is displayed, listing all active systems of the Sysplex as stored in the sysplex couple data set of the system.
2. After selecting the system to be verified a context menu with two verification actions is displayed. Select the *Verify active configuration against system* task (action code **k**) to compare the system against the active IODF. Select the *Verify target configuration against system* task (action code **1**) to compare the system against the accessed IODF.

The Identify System I/O Configuration panel is displayed (see Figure 100).

3. Specify the processor ID and OS configuration ID. If the specified processor is in LPAR mode, you must also enter a partition name.

The I/O path list

The I/O Path List is the output of the verify function available on the Active Sysplex Member List and the Activate or Verify Configuration Dynamically panel (see “Verifying a configuration against the local system” on page 228 “Verifying a configuration against a system in the sysplex” on page 228). The list compares the configuration in the accessed or the active IODF with the actual configuration as sensed from the system.

The report extends over two pages and can be scrolled horizontally.

```

Goto Filter Backup Query Help
-----
I/O Path List                               Row 1 of 473 More: >
Command ==>>                               Scroll ==>> PAGE

Accessed IODF: IODFST.IODF11
Active IODF  : IODFST.IODF11                System . : CB88
Processor   . : CB88           Partition :    OS config: B710

----I/O Path-----Sensed Data-----IODF Data-----D
CHP CU  DEV  STAT CHT  CUTYPE  DEVTYPE  O CHT  CUTYPE  DEVTYPE  S
10 FFFE FFFE,2  CFS  9674-C04 CFS      CFS CFS  CFS      C
11 FFFD FFFC,2  CFS  9674-C04 CFS      CFS CFS  CFS      C
20 0040 0040,32 CVC_P ****  ****      Y CNC 3990 3380  *
21 EFD2 EFD2,4  CNC_S 9672-E02 9672-CTC CNC SCTC  SCTC
21 EFD2 EFD6   CNC_S 9672-E02 9672-CTB CNC SCTC  BCTC
21 EFE2 EFE2,4  CNC_S 9672-E02 9672-CTC CNC SCTC  SCTC
22 0A00 0A00,64 CNC_S ****  ****      Y CNC 3990 3380  *
22 0A80 0A80,2  CNC_S 3990-6 3390-A38 Y CNC 3990 3390
22 0A80 0A82   CNC_S 3990-6 3390-A38 CNC 3990 3390
22 0A80 0A83   CNC_S 3990-6 3390-A38 Y CNC 3990 3390
22 0A80 0A84   CNC_S 3990-6 3390-A38 CNC 3990 3390
30 2000 202F   6 S FC_S 2107-9A2 2107-900 FC 2107 3390B  *
30 2000 2030,32 FC_S 2107-9A2 2107-900 Y FC 2107 3390B  *
B5 OFFL OSM
FC OFFL IQD
F1=Help F2=Split F3=Exit F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel F20=Right F22=Command

```

For each channel path sensed and/or defined in the accessed IODE, the list contains a row showing the I/O path and the sensed and defined channel path, control unit, and device information. If a switch is included in the path, the right page shows the corresponding switch information.

Any discrepancies between the defined and the sensed data are indicated in column D on the right. For channel paths for which column D is blank, the defined and sensed data are consistent. The following values may appear:

Symbol

Meaning

- * Defined and sensed I/O paths differ
- C Defined only to processor but not to OS
- O Defined only to OS but without a path to the processor
- @ A combination of * and C

On the display column D is highlighted.

Columns STAT and O indicate the status of the I/O path and the connected device, respectively. An empty field means that the corresponding I/O path or device is online. Offline I/O paths are marked with OFFL and offline devices with Y. If the

system is unable to sense the status of an I/O path, it is marked UNKN. If a path is online, HCD checks, whether a single point of failure (SPOF) can be found for this path. If yes, this single point of failure is mapped to the following

- BLANK: Sensed I/O Path is online and no SPOF exists
If the sensed I/O path is online and a SPOF exists, an up to four digit number is shown with a SPOF indication as follows:
- Position 1 contains blank or SPOF indication 3 (book), 4 (cage), 5 (fail-over domain), 6 (fan-out), 7 (domain), 8 (secondary STI/STI).
- Position 2 contains C if the controller interface shares a SPOF.
- Position 3 contains P if the device has only one path online.
- Position 4 contains S if all paths go through the same switch.

Asterisks (*) in the CHT (channel type), CUTYPE (control unit type), or DEVTYPE (device type) columns indicate that I/O paths are returned but the values for the corresponding types are blank or invalid.

For certain configurations the I/O path list, although restricted to one processor or partition, can be extensive. Using the Filter action you can reduce the list to the entries of interest.

You can save the displayed list by entering SAVE in the command line on the I/O Path List.

Note: The LOCATE command is not available for the I/O path list.

Activate a configuration dynamically

The system programmer (or other authorized persons) can use the option *Activate or verify configuration dynamically* or the ACTIVATE operator command to make changes to a running configuration. That is, the possibility is offered to change from a currently active configuration to some other configuration that is to be made active without the need to POR or IPL the system again.

When activating a configuration dynamically, HCD compares the currently active IODF with the IODF that is to be activated and then processes the difference.

For the IODF that is to be activated, HCD uses the production IODF that is currently in use with the dialog. Use the same high-level qualifier for the currently active IODF and the IODF to be activated.

z/OS HCD Planning gives a detailed description of how to dynamically activate a configuration. It describes the prerequisites for a dynamic activation, explains when hardware and software changes or software-only changes are allowed, and describes the actions necessary to change your I/O configuration dynamically. The following sections describe how to use the HCD dialog for this purpose.

Before activating a configuration dynamically, you may want to view information about the IODF that has been used for IPL or the last dynamic activation.

View active configuration

HCD allows you to view the name and status of the IODF that has been used for IPL or for last dynamic activation. The operating system configuration and EDT identifier and, if applicable, the configuration token, which is currently active in the HSA (hardware system area), are shown. Use the *View active configuration*

function for an overview of the actual status for dynamic activation, indicating whether hardware and software changes are allowed.

1. On the *Primary Task Selection* panel, select *Activate or process configuration data* and then *View active configuration*.

The *View Active Configuration* window with sample data is shown:

```

View Active Configuration

Currently active IODF . . . : SYS1.IODF01
Creation date . . . . . : 1995-08-18
Volume serial number . . : SYSPAG

Configuration ID . . . . . : MVSVM          MVS Testsystem on VM
EDT ID . . . . . : 00

HSA token . . . . . : LMIF9672 96-08-02 13:22:50 SYS1      IODF50

Activation scope:
Hardware changes allowed .: Yes
Software changes allowed .: Yes

ENTER to view details on the activation scope.

```

2. Press the Enter key to display detailed information on limitation(s) to the activation scope. A *Message List* is displayed containing the messages about the reasons for the restrictions.

How to activate if hardware and software changes are allowed

The following procedure describes how to activate a configuration dynamically if both hardware and software changes are allowed. Refer to *z/OS HCD Planning* on information when both hardware and software configuration changes are allowed and when only software configuration changes are allowed.

1. On the *Primary Task Selection* panel, select *Activate or process configuration data*, and from the resulting panel select *Activate or verify configuration dynamically*. HCD displays the *Activate or Verify Configuration* panel.

```

Activate or Verify Configuration

The currently active IODF matches the hardware I/O
configuration. Both hardware and software definitions may be
changed. Select one of the following tasks.

— 1. Activate new hardware and software configuration.
   2. Activate software configuration only. Validate
      hardware changes. Process changes to Coupling
      Facility elements.
   3. Activate software configuration only.
   4. Verify active configuration against system.
   5. Verify target configuration against system.
   6. Build CONFIGxx member.

F1=Help   F2=Split   F3=Exit   F9=Swap   F12=Cancel

```

2. Select what you want to activate. The following figure assumes that you selected task 1. Activate new hardware and software configuration. The panels when you select the other tasks are similar.

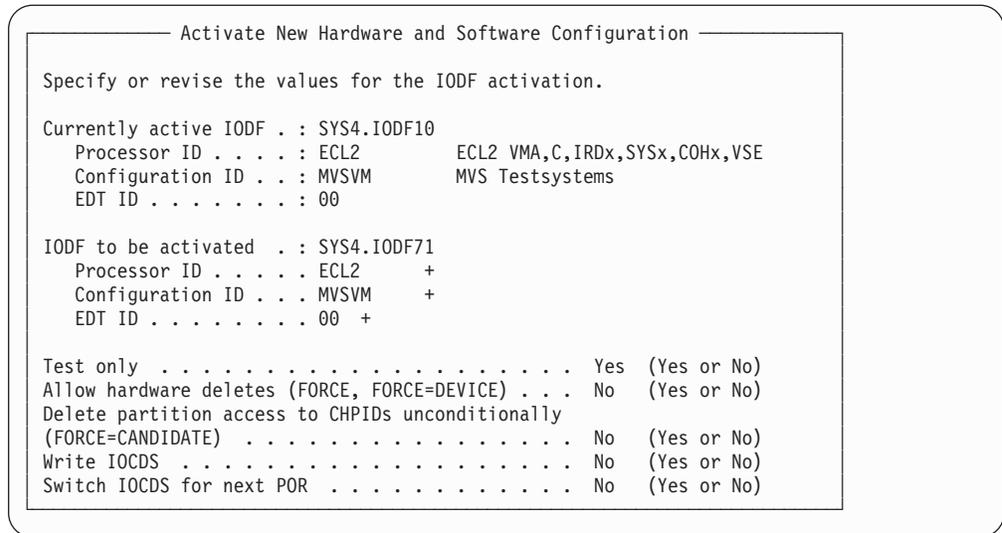


Figure 101. Activate New Configuration: Hardware and Software Changes

The panel contains information about the currently active IODF.

3. On the Activate New Hardware and Software Configuration panel you can change the fields that relate to the IODF that is to be activated, and you can specify options as applicable to your requirements. It is recommended that you first specify to test an activation before you dynamically activate a configuration.

Allow hardware deletes option:

If logical partitions have been defined for the currently active configuration, you can specify whether you want to allow hardware deletes.

Yes means that the hardware deletes become effective for all partitions.

No (the default) means that, if the changes include requests for deleting hardware, the activation is rejected.

Note that hardware delete can also be indirectly performed as a result of other changes, for example, a change of a channel path consist of a deletion and an addition of a channel.

A configuration change is rejected if it includes a hardware delete for an I/O component that is online to the logical partition from which you are making the change, even if you have entered Yes in the Allow hardware deletes option field. Therefore, you should vary offline any affected I/O component in all logical partitions. For example, when changing a channel path from unshared to shared, you must allow hardware deletes, and you must configure the channel path offline and vary offline the associated I/O devices before you activate the configuration. See *z/OS HCD Planning* for details about preventing disruption when changing the characteristics of I/O components.

Delete partition access to CHPIDs unconditionally option:

You can also specify how this activation should treat any deletion of a partition from the access or candidate list of a channel path. In the field Delete partition access to CHPIDs unconditionally (FORCE=CANDIDATE), enter either Yes or No. If you specify Yes, the access to the channel path will be revoked even if the channel is configured online to the partition; the channel will be configured off-line to the partition, and active I/O operations might be disrupted. If you specify No (the default), the activation will be rejected if it includes a deletion of partition access to a channel path that is configured online to that partition.

Note: You cannot unconditionally delete the partition that is invoking the activate request from the candidate or access list of a channel path if the channel path is currently configured online.

4. If the dynamic activation completed successfully, HCD displays a message.

Configure channel path online to the partition

When activating a configuration in which a partition is added to the access list of a channel path, you must configure the channel path online to the partition using either the MVS CONFIG command or the processor console CHPID command. PR/SM will configure the channel path to the partition at subsequent activations only after you configure it using the previous commands at least once.

When a particular IOCDS is used in a POR for the first time after it has been written, the definitions in that IOCDS are used to determine the assignment of channel paths to logical partitions according to the channel path access lists that are defined. All previous information about channel configurations associated with this IOCDS is discarded. The exception to this rule is when a newly written IOCDS is first used as part of a dynamic I/O change to the system. For example, the new IOCDS is used as a result of a *Switch IOCDS for next POR* or the new IOCDS is the target of the ACTIOCDS= parameter of the MVS ACTIVATE command. When a new IOCDS is used in this manner, the current state of channel configurations is preserved and immediately associated with the newly written IOCDS.

See *z/OS MVS System Commands* for information about the MVS CONFIG command and see the applicable *Processor Resource/Systems Manager Planning Guide* for information about the CHPID command and for a description of automatic configuration of channel paths to partitions.

If dynamic activation fails

If the activation is rejected, HCD displays a panel that lists the messages and reasons for a failure. From the displayed message list, you can request further information. The message list can also be displayed by using the command SHOWMSG ACTIVATE on any panel that has a command line (except on help panels).

In some cases a dynamic activation may fail and HCD recommends recovery. HCD displays a panel where you can specify whether you want to recover:

- If you confirm recovery by specifying Yes, HCD performs hardware-only changes. You can specify to recover in two ways:
 - To resume activation of the target IODF. That is, HCD tries to continue with the activation.
 - To reset the configuration to the source IODF. That is, HCD activates the configuration that existed before the failure occurred.
- If you do not confirm recovery by specifying No, HCD allows you to continue with software-only changes. Hardware changes are activated up to the point where the failure occurred.

Detection of illegal split/merge of LCU: If a request for activating a new configuration causes a logical control unit (LCU) to be split or merged illegally, HCD rejects the request. HCD considers an LCU to be:

- Illegally split if a physical control unit (PCU) is removed from the LCU that has devices remaining connected to it.
- Illegally merged if a physical control unit is added to an LCU that has devices connected to it.

HCD detects every split/merge during activation and informs you by message. The message tells between which PCU and which device the split/merge occurred; it also explains how to correct the condition.

For more information on how to recover after a system failure, refer to *z/OS HCD Planning*.

How to activate if software-only changes are allowed

The following procedure describes how to activate a configuration dynamically if only software changes are allowed. Refer to *z/OS HCD Planning* on information when both hardware and software configuration changes are allowed and when only software configuration changes are allowed.

1. On the *Primary Task Selection* panel, select *Activate or process configuration data*, and from the resulting panel select *Activate configuration dynamically*. HCD displays the following panel:

```

----- Activate Software Configuration Only -----
Specify or revise the values for IODF activation.

Hardware changes will be validated, and the activation will be rejected
in the case of an error.

Currently active IODF . : SYS4.IODF10
Processor ID . . . . . : ECL2           ECL2 VMA,C,IRDx,SYSx,COHx,VSE
Configuration ID . . . : MVSVM         MVS Testsystems
EDT ID . . . . . : 00

IODF to be activated . : SYS4.IODF71
Processor ID . . . . . : ECL2         +
Configuration ID . . . : MVSVM         +
EDT ID . . . . . : 00                 +

Test only . . . . . : Yes           (Yes or No)
```

The panel contains information about the currently active IODF.

2. On the Activate Software Configuration Only panel, you can change the fields that relate to the IODF that is to be activated. It is recommended that you first test an activation before you actually dynamically activate a configuration.
3. If the dynamic activation completed successfully, HCD displays a message.

If dynamic activation fails

If the activation is rejected, HCD displays a panel that lists the message(s) and reasons for a failure. From the displayed message list, you can request further information. The message list can also be displayed by using the command `SHOWMSG ACTIVATE` on any panel that has a command line (except on help panels).

Activate a configuration sysplex-wide

HCD offers you a single point of control for systems in a sysplex. You can now dynamically activate the hardware and software configuration changes for each system in a sysplex from any other system in the same sysplex. You can:

- Display active sysplex members
- Activate Software Configuration Changes Only

- Activate Software and Hardware Configuration Changes
- Switch IOCDS for the next POR

z/OS HCD Planning gives a detailed description of how to dynamically activate a configuration. It describes the prerequisites for a dynamic activation, explains when hardware and software changes or software-only changes are allowed, and describes the actions necessary to change your I/O configuration dynamically. The following sections describe how to use the HCD dialog for dynamically activating systems in a sysplex.

Displaying active sysplex members

Before you can make any change to a configuration in a sysplex, you must display the Active Sysplex Member List. From this list you then select different actions.

1. On the *Primary Task Selection* panel, select *Activate or process configuration data*, and from the resulting panel select *Activate configuration sysplex-wide*. HCD displays the Active Sysplex Member List.

```

Goto Query Help
-----
Active Sysplex Member List                               Row 1 of 3
Command ==> _____ Scroll ==> PAGE

Select one or more systems, then press Enter. To refresh the Activate/Verify
Status, press Enter without selections made.

IODF to be activated: SYS4.IODF71
Active sysplex . . . : IRD4PLEX

 System  Processor Partition Active      Config. EDT Act./Verify
/ Name   ID       Name   IODF      ID    ID  Status
- IRD4   ECL2    IRD4   SYS4.IODF10 MVSVM 00
- IRD5   ECL2    IRD5   SYS4.IODF10 MVSVM 00
- IRD6   R35     IRD6   SYS4.IODF10 MVSVM 00
***** Bottom of data *****

```

Figure 102. Activate Sysplex Member List

You can see the system names, and the processor IDs and partition names associated with the system names. You can also see the IODF to be activated, the name of the sysplex, the active IODFs, the configuration IDs and EDT IDs used for IPL, and the Activate status, which is empty initially.

HCD requests the information from the sysplex couple data set and the HSA of every CPC and displays it in a formatted list.

Refreshing the Active Sysplex Member List: The Active Sysplex Member List will be refreshed whenever you press the Enter key. If a system joins the sysplex, it will be added to the list in alphabetical order. If a system leaves the sysplex, it will be deleted from the list.

Empty Processor ID: The entry in the Processor ID column is empty when the operating system runs as a VM guest.

This does not mean that you cannot initiate an activation request for that system. It means that you have to specify the correct processor ID later on by yourself, as required.

2. Select a system name and specify *View the configuration status* from the context menu (or select action code **v**) to see the following information:
 - information about the currently active hardware configuration token stored in the HSA
 - information about the free space in the HSA.

Activate software configuration changes only

The following procedure describes how to change the software configuration for one or more systems in a sysplex.

1. Select one or more systems from the Active Sysplex Member List (see Figure 102 on page 235) and the *Activate software configuration only* action from the context menu (or action code **o**). The *Activate Software Configuration Only* dialog is displayed.

Activate Software Configuration Only Row 1 of 3

Specify or revise the values for activation, then press Enter.

IODF to be activated: SYS4.IODF71

System Name	Processor ID +	Partition Name	Config. ID +	EDT ID +	Valid. HW Ch.	Test Only	Active IODF
IRD4	ECL2	IRD4	MVSVM	00	Yes	Yes	SYS4.IODF10
IRD5	ECL2	IRD5	MVSVM	00	Yes	Yes	SYS4.IODF10
IRD6	R35	IRD6	MVSVM	00	Yes	Yes	SYS4.IODF10

***** Bottom of data *****

Figure 103. Activate Software Configuration Only

All systems you have selected are shown together with the associated processor IDs.

Empty Processor ID: If the *Processor ID* field is empty, prompt for the processor IDs and select the actual one.

Empty Configuration or EDT ID: If the *Config. ID* and the *EDT ID* fields are empty, it is an indication that the ID of the currently active configuration is not defined in the IODF to be activated. Prompt for the new ID. Updating the *processor ID*, the *Config. ID* or the *EDT ID* fields might be required if your IODF to be activated contains IDs different from those displayed as default IDs.

2. On the Activate Software Configuration Only panel, update the fields of one or more systems. It is recommended that you first test an activation before you actually dynamically activate a configuration.
3. If the hardware token matches, the *Valid. HW Ch.* option is set to 'Yes'. It is recommended to validate hardware changes when performing a software change. This is required when the configuration change contains coupling facility control units or devices.
4. After pressing the Enter key, the Active Sysplex Member List is displayed again, but now the Active Status column shows the status *In progress*. If you refresh the list occasionally, you can see that one system after the other completes the activation request. This is indicated by the status *Messages*.
5. Select a system and the *View messages* action from the context menu (or action code **m**). The messages returned from that system as the result of the activation request are displayed.

```

----- Message List -----
Save Query Help
-----
Row 1 of 4

Select one or more messages, then press Enter. To refresh the list, press
ENTER without selections made. To reply to a WTOR message, use the REPLY
command.

System name: SC52

/ Message Text
_ IOS500I ACTIVATE RESULTS 084
# ACTIVATE COMPLETED SUCCESSFULLY
# NOTE = 0100,SOFTWARE-ONLY CHANGE
# COMPID=SC1C3
***** Bottom of data *****

```

Figure 104. Message panel

- If you do not need the messages any longer, you can delete them by using the option *Delete messages* (or action code **d**).

If dynamic activation fails

If the activation is rejected, HCD displays a panel that lists the messages and reasons for a failure.

To get more information you also use the *View configuration status* option (or action code **v**). This option provides you the same information as when entering the command D IOS,CONFIG(ALL) at the system console.

```

----- Message List -----
Save Query Help
-----
Row 1 of 12

View configuration status of selected systems.

Message Text
IOS506I 06.49.55 I/O CONFIG DATA 378
ACTIVE IODF DATA SET = SYS6.IODF29
CONFIGURATION ID = L06RMVS1      EDT ID = 01
TOKEN: PROCESSOR DATE      TIME      DESCRIPTION
SOURCE: P101  95-05-09 15:34:56 SYS6   IODF29
      48 PHYSICAL CONTROL UNITS
      547 SUBCHANNELS FOR SHARED CHANNEL PATHS
      548 SUBCHANNELS FOR UNSHARED CHANNEL PATHS
      23 LOGICAL CONTROL UNITS FOR SHARED CHANNEL PATHS
      23 LOGICAL CONTROL UNITS FOR UNSHARED CHANNEL
      PATHS

```

Figure 105. View Configuration Status

HCD offers you two possibilities after an activation failed:

- Select *Resume activation of target configuration* (or action code **t**) to force the activation of the system.
- Select *Reset source configuration* (or action code **r**) to reset the original configuration.

Activate software and hardware configuration changes

The following procedure describes how to change the software and hardware configuration for one or more systems in a sysplex.

1. Select one or more systems from the Active Sysplex Member List and the *Activate software and hardware configuration* action from the context menu (or action code **a**). The Activate Hardware and Software Configuration panel is displayed.

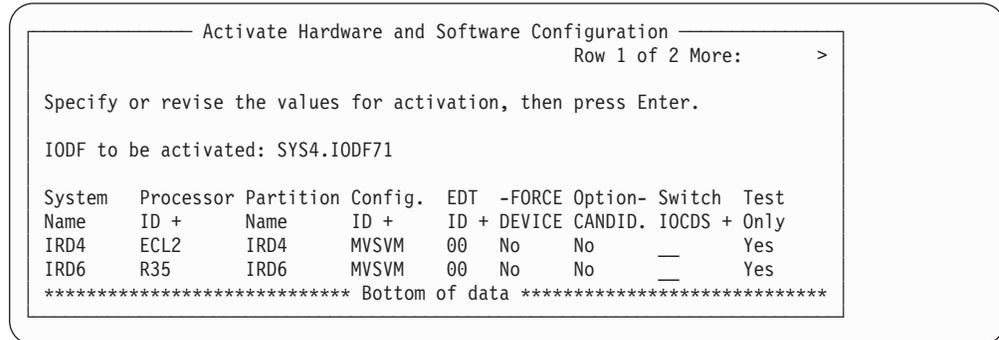


Figure 106. Activate Hardware and Software Configuration

The systems you have selected are shown together with the associated processor IDs.

Empty Processor ID:

If the Processor ID field is empty, prompt for the processor IDs and select the actual one.

Empty Configuration or EDT ID:

If the Config. ID and the EDT ID fields are empty, it is an indication that the ID of the currently active configuration is not defined in the IODF to be activated. Prompt for the new ID. Updating the Processor ID, the Config. ID or the EDT ID fields might be required if your IODF to be activated contains IDs different from those displayed as default IDs.

Switch IOCDS:

In the Switch IOCDS column, you can define the IOCDS name used for the next POR.

Note: If you prompt for IOCDS, you might see outdated IOCDS update dates, if it was not possible with the last write IOCDS action to update the IOCDS records in the IODF. To ensure update of the IOCDS dates, use option 2.2 (write IOCDS) for the processor in focus.

2. On the Activate Hardware and Software Configuration panel, update the fields of one or more systems. It is recommended that you first test an activation before you dynamically activate a configuration.

If you plan to delete a device for a specific system, specify *Yes* in the *FORCE DEVICE* field of that system. If you plan to remove a partition from the access or candidate list of a channel path belonging to a specific system, specify *Yes* in the *FORCE CANDID.* field of that system.

3. After updating the panel, press the Enter key. The Active Sysplex Member list will be displayed again, but now the *Active IODF* fields have changed for the affected systems and now contain the name of the IODF to be activated. If you have defined different processor IDs, configuration IDs or EDT IDs, and you have activated their configurations, these fields also have been changed.

4. Select a system and the *View messages* action from the context menu (or action code **m**). The messages returned from that system as the result of the activation request are displayed.
5. If you do not need the messages any longer, you can delete them by using the option *Delete messages* (or action code **d**).

If Dynamic Activation Fails

Refer to “If dynamic activation fails” on page 237 for information on how to proceed, if your activation has been rejected.

Switch IOCDS for the next POR

In addition to dynamically activating a sysplex, you can also specify the IOCDSs to be used for the next POR from the Active Sysplex Member List.

Refer to “Switch IOCDS for systems in a sysplex” on page 250 for a description of how to switch the IOCDS for the next POR.

Activate a configuration HMC-wide

HCD provides a function to manage dynamic I/O changes from a single point of control (managing system) for all systems running on CPCs that are controlled by the same hardware management console (HMC). The function includes the following features:

- listing all partitions on the HMC-controlled CPCs with their status (deactivated, activated, IPLed), and partition dependent information
- distributing a production IODF to selected target systems
- remotely performing dynamic hardware or software changes, or both, on selected target systems and showing the resulting messages
- operating selected target systems for the activation remotely and showing the resulting messages

To work with HMC-wide activate functions, start from the *z Systems Cluster List* (Figure 96 on page 215), where you can select action *Work with CPC images* (action code **v**) to obtain the *CPC Image List* (see Figure 107 on page 241).

The *CPC Image List* displays information about the selected CPC. This includes the processor ID, SNA address, and the HSA token.

Partitions that are deactivated or not IPLed, or that do not run any z/OS (MVS) or z/VM (VM) system are shown as disabled (with a #sign in the action column) and no action is possible for such a partition.

Also all LPARs without an entry in the connection table which are not reachable indirectly via a reachable sysplex are listed but marked as disabled (#) in the action column.

The *CPC Image List* displays all partitions defined on the selected CPC with their operation status, connection status, and activation status:

- partition name
- partition ID (LPID from the support element)
- operation status of the partition:
 - 0001** The partition is operational and a system is IPLed.
 - 0002** The partition is activated but no system is IPLed.
 - 0008** The partition is deactivated.

0010 There are exceptions.

Any other value indicates a special condition for the partition that you can check at the Support Element (SE) of the selected CPC or the HMC.

- The connection status (column *Co ST*) for z/OS and z/VM systems running in the image, indicating whether this system is reachable or not. This information depends on the content of your connection table.

BLANK Indicates that no entry for the system exists in the connection table and therefore, a connection between HCD of the local system to the displayed system does not exist.

Y Indicates that a connection between HCD of the local system to the displayed system exists because of an entry in the connection table.

S Indicates that a connection between HCD of the local system to the displayed system exists because they are either identical or in the same sysplex, or another member of the sysplex has a direct connection via an entry in the connection table.

N Indicates that a connection between the local HCD and the displayed system failed in spite of an entry in the connection table.

- The activation status of the image contains information about its activation capabilities and currently active configurations.

Activation HW

Indicates whether a Hardware Activate is possible: Y or not: N, or whether it is unknown: ?.

Activation SW

Indicates whether a Software Activate is possible (Y or N). If the software is not managed by HCD, a switch of the IODF is possible: S.

Activation Status

Indicates the status of an activate request for connected z/OS and z/VM systems (for example, *Activating* or *In Progress*). It indicates also whether hardware or software changes or both are possible.

```

Goto Query Help
-----
CPC Image List          Row 1 of 60 More:  >
Command ==> _____ Scroll ==> PAGE

Select one or more systems of the CPC, then press Enter. To refresh the
Activate/Verify Status, press Enter without selections made.

Processor ID : R35          SNA Address : IBM390PS.R35
HSA Token   . : R35      12-02-06 09:26:03 SYS4      IODF71

  -- Partition --- Co Sysplex  ----- System -----  ----- Activation -----
/ Name      ID Stat St Name      Name      Type  Level  HW SW Status
- DWB1      2A 0001  DWB1PLEX DWB1      MVS    1.12
- DWB2      3A 0001  DWB1PLEX DWB2      MVS    1.12
- IRD6      1B 0001  Y  IRD4PLEX IRD6      MVS    2.1    Y  Y
- R35LP29   1E 0001  R35SSI   BOER3529 VM     6.2
- R35LP43   2D 0001  R35SSI   BOER3543 VM     6.2
- TRX1      2C 0001  S  TRX1PLEX TRX1      MVS    2.1    Y  Y
- TRX2      3C 0001  Y  TRX1PLEX TRX2      MVS    1.13   Y  Y
# IRD7      0C 0002
# IRD8      1C 0002  N
# R35LP10   0A 0001                LINUX    3.0.13

F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F10=Actions  F12=Cancel   F13=Instruct F14=SortName F15=SortStat F16=SortSplx
F22=Command

```

Figure 107. CPC Image List

Scrolling to the right displays further activation status information as shown in Figure 108 on page 242:

- active IODF
- active OS configuration ID in column **Conf ID** (available for z/OS systems; available for z/VM systems only if z/VM manages the software configuration via HCD)
- active EDT ID (only available for z/OS systems)

```

Goto Query Help
-----
CPC Image List          Row 1 of 60 More: <
Command ==> _____ Scroll ==> PAGE

Select one or more systems of the CPC, then press Enter. To refresh the
Activate/Verify Status, press Enter without selections made.

Processor ID : R35          SNA Address : IBM390PS.R35
HSA Token   . : R35       12-02-06 09:26:03 SYS4   IODF71

  Partition Active          Conf      EDT
 / Name      IODF          ID        ID
- DWB1
- DWB2
- IRD6      SYS4.IODF10      MVSVM     00
- R35LP29
- R35LP43
- TRX1      SYS1.IODF01      MVSVM     00
- TRX2      SYS1.IODF01      MVSVM     00
# IRD7
# IRD8
# R35LP10

F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F10=Actions  F12=Cancel   F13=Instruct F14=SortName F15=SortStat F16=SortSp1x
F22=Command

```

Figure 108. CPC Image List scrolled to the right

Use PF keys to sort the list by the following columns:

- F14** sort by *Partition Name*
- F15** sort by *Partition Stat (operational status)*
- F16** sort by *Sysplex Name*

From the *CPC Image List* you can launch dynamic hardware and software configuration changes on any reachable system (in any sysplex) on any CPC that is configured in the same Hardware Management Console (HMC). All available actions are introduced in “Available configuration activation actions.”

Press the Enter key without further selection to refresh the display of the activation status, for example, when there is an activation in progress.

The save list function is available for the *CPC Image List*.

Note: If you stopped and restarted an HCD dispatcher on a connected system, while you have opened a *CPC Image List* connected to the CPC where this system is running, you must reconnect this CPC before performing any new action on the *CPC Image List* containing the concerned system. Otherwise you may get message CBDA605I, saying that HCD remote processing failed.

Available configuration activation actions

The following configuration actions are available from the *CPC Image List*:

- *Activate software configuration only* (action code **o**). This action is available if column *Activation SW* shows Y or S.
- *Activate software and hardware configuration* (action code **a**). This action is available if column *Activation HW* shows Y.
- *Resume activation of target configuration* (action code **t**). This action is needed if a previous hardware activation failed and the target system requires recovery. Column *Activation HW* must show Y.

- *Reset source configuration* (action code **r**). This action is needed if a previous hardware activation failed and the source system requires recovery. Column *Activation HW* must show Y.
- *Transmit IODF* (action code **x**). The Export IODF dialog is invoked. This action is available for a selectable image.
- *Process system commands* at the target system and show the result of the command execution on the managing system (action code **c**).

The actions *Resume activation of target configuration* and *Reset source configuration* use the same dialogs, interaction and processing as the corresponding actions available from the *Active Sysplex Member List* as described in “Activate a configuration sysplex-wide” on page 234.

Action *Transmit IODF* allows sending an IODF to a target system. This task does not require that a connection exists between the local HCD client (managing system) and the HCD of the target system.

For details about the remaining actions, see the following information units:

- “How HCD processes Activate actions”
- “How HCD processes system commands” on page 244

How HCD processes Activate actions

With action codes **o** (for *Activate software configuration only*) and **a** (for *Activate software and hardware configuration*), you can send the corresponding activation requests to the selected remote target systems. The invoked dialog and the processing is the same as with actions available from the *Active Sysplex Member List* as described in “Activate a configuration sysplex-wide” on page 234. However, the request is sent to the selected target system and the messages are returned from that system to the managing system.

When performing an Activate action on a remote system, HCD checks whether the currently accessed target IODF is already available at the remote system. HCD looks for a matching IODF name with a high-level qualifier that corresponds to the high-level qualifier (HLQ) of the active IODF on the remote system. If there exists one, HCD checks whether the IODF token of that IODF and the IODF token of the target IODF match. If they do match, the new target IODF can be accessed by the remote system and the remote Activate request can be sent. For example, if the target IODF on the managing system is J0E.I0DFA1 with an IODF token of token777, and the HLQ of the currently active IODF on the managed system is SYS4, then HCD looks for IODF SYS4.I0DFA1. If this IODF is found, and has the IODF token token777, this existing IODF is used for the remote activate request.

If the target IODF can not be accessed from the remote system due to failing conditions, the target production IODF is transferred to the remote system cataloged there with an HLQ that corresponds the HLQ of the active IODF on the target system and on the same volume as the active IODF resides on. Before a production IODF on the remote system is replaced, you must confirm the deletion of the old IODF version. Data sets that are associated with the production IODF, like activity log, change log or HCM MCF data set are not sent.

If the target system is an MVS system, the remote system processes the Activate request in the same way as a sysplex-wide activate action that is issued on the target system. This means, an ACTIVATE system command is built and routed to the corresponding system in the sysplex. Activate processing is done asynchronously. On the *CPC Image List* on the local (managing) system, the field *Activation Status* is changed to Activating. You can refresh the status by pressing

Enter without selecting any partition. If activate processing has not yet completed, the *Activation Status* field shows In progress. If the activate processing has completed on the remote system, this field switches to Messages to indicate that the resulting messages have been received from the remote system.

If the target system is a VM system, the remote system will process the Activate request synchronously. The local system waits until activate processing has been completed on the remote system. The *Activation Status* field shows Activating. Press Enter to switch the activation status to Messages immediately.

Now you can view the resulting messages by selecting action *View messages* (action code **m**). These messages indicate how the activation request has processed on the remote system(s).

HCD allows you to enter activate requests to multiple partitions at a time. The requests for systems of the same sysplex are performed as a single remote activate request.

How HCD processes system commands

Applying action code **c** on a selected system in the *CPC Image List* invokes the *Process System Command* dialog. Here you can enter a system command that you want to be processed on the target system or on all systems belonging to the same sysplex. The sample dialog in Figure 109 shows the entered system command `d ios,config`.

The entered command must either be an MVS system command, if the target system is of type MVS, or a CP command, if the target system is of type VM. For target VM systems, command routing to other systems in the sysplex is not supported. However, HCD does not perform any verification or validation on the command syntax, but sends the entered string to the target system(s) to be executed there.

```
Process System Command

Specify a system command. It will be executed on the selected target
system. Press ENTER to continue.

System . . : TRX1
Sysplex . . : TRX1PLEX

Command . . d ios,config_____

Do you want the specified command being
executed on all systems in the sysplex ?  1  1. Yes
                                           2. No
```

Figure 109. Process System Command

When you press Enter, HCD redisplay the *CPC Image List* where column *Activation Status* now shows Executing for all systems where the command is processed. When you press Enter again, the *Activation Status* changes to Messages.

Now select action *View messages* (action code **m**) to display the message(s) resulting from the system command as it had processed on the remote system. An example of a returned message list for the `d ios,config` command is shown in

Figure 110.

```

----- Message List -----
Save Query Help
-----
Row 1 of 8
Command ==> _____ Scroll ==> PAGE

Select one or more messages, then press Enter. To refresh the list, press
ENTER without selections made. To reply to a WTOR message, use the REPLY
command.

System name: TRX1

/ Message Text
- IOS506I 17.59.32 I/O CONFIG DATA 776
# ACTIVE IODF DATA SET = SYS1.IODF01
# CONFIGURATION ID = MVSVM          EDT ID = 00
# TOKEN: PROCESSOR DATE      TIME      DESCRIPTION
# SOURCE: R35      11-12-13 09:42:28 SYS4      IODF71
# ACTIVE CSS: 2      SUBCHANNEL SETS CONFIGURED: 0, 1, 2
# CHANNEL MEASUREMENT BLOCK FACILITY IS ACTIVE
# LOCAL SYSTEM NAME (LSYSTEM): R35
***** Bottom of data *****

```

Figure 110. Process System Command - returned messages

Establishing connectivity to remote systems

To establish connectivity to remote systems, you must provide connection data in a connection table. The data set containing the connection table is defined to HCD via the profile option CONNECTION_TABLE as described in “HMC-wide activation” on page 29. Specify the following information in the connection table:

- Processor SNA address with network name and CPC name as defined for the corresponding processors in the IODF and as known in the HMC/SE.
- Partition (image) name as specified in the IODF and configured in the SE.
- IP address or host symbolic destination name for the TCP/IP target system where the HCD dispatcher program is running.
- IP port ID which is used by the HCD dispatcher program at the remote site.
- User ID for the remote system.
- Optional: Password for user ID on the remote system. If you do not provide a password, HCD uses a PassTicket for verifying the authorization for the user ID on the remote system. In this case, you must provide the corresponding RACF definitions (see “How to set up PassTickets for working with CPC images on z/OS” on page 362 and “How to set up PassTickets for working with CPC images on z/VM” on page 364).

Specify the values in a single line separated by commas. You can insert comment lines using an * in column 1.

Example:

```

***** Top of Data ****
* NETWORK NAME      IMAGE      IP ADDR  PORT  USERID  PASSWORD
*-----
IBM390PS,R35      ,TRX2      ,BOETRX2 ,51107,USERID10,PASSWRD1
IBM390PS,R35      ,TRX1      ,BOETRX1 ,51107,USERID10,PASSWRD1
IBM390PS,DAN2      ,SYSB      ,BOESYSB ,51107,USERID10,PASSWRD1
*BM390PS,DAN2      ,SYSA      ,BOESYSA ,51107,USERID20,PASSWRD2
IBM390PS,POL1      ,HCDVM     ,BOEHCDVM,51107,USERID20,PASSWRD2
***** Bottom of Data **

```

To be able to perform actions on a remote z/OS system (type MVS), a connection must exist to at least one system of the sysplex where the remote system is part of. For a remote z/VM system (type VM), a connection must exist to that system. On the remote systems, the HCD dispatcher program for remote API calls must be running and listening to the specified ports.

HCD tries to establish an initial connection for all specified target systems in the connection table when the first *CPC Image List* is invoked. If a connection can be established and the remote HCD program is running, the connection status shows a Y. If the initial connection failed, the connection status shows N.

Note: The *CPC Image List* is not refreshed as long as the underlying *z Systems Cluster List* is open. To refresh, leave the *z Systems Cluster List* and reopen the *CPC Image List*.

Connections to z/OS or z/VM systems that are not supported (z/OS before V1.10 or z/VM before V5.4) are rejected with an error message.

The connection table data set may be sequential or a member of a partitioned data set. Record format of the data set must be F or FB with a logical record size of at least 80 characters. If no connection table is specified, the *CPC Image List* is displayed, but no actions are available for the listed images (with the # sign in the action column).

Prerequisites for working with CPC images

- HCD uses BCPii (Base Control Program internal interface) to query the image (partition) attributes from the Support Element (SE) of the selected CPC. In order to get the partition attributes, the following requirements must be met:
 - The BCPii address space (HWIBCPii) is active and ready to handle BCPii requests.
 - The local and remote support elements (SEs) are enabled for BCPii communication (cross-partition authority must be enabled for each CPC that is queried).
 - The BCPii community name must be defined on the SE for the local and each remote CPC that is queried.
 - The managing user ID must get the authorization to perform the BCPii calls for the target CPCs / images.
- The target systems must have a TCP/IP connection to the managing system. For more information, refer to Appendix E, “Establishing the host communication,” on page 505.
- The user must provide a connection table for the target systems which contains the TCP/IP login data.
- There must be a user ID on each target system that has the authorization to perform dynamic activations and corresponding system commands and that has access to the active production IODF.
- An HCD dispatcher must run on each of the target systems that allows directing incoming remote HCD requests to the local HCD versions.
- The required security setup for using this function is described in “Defining RACF profiles” on page 359 and “Access to HWI.* profiles” on page 362.

Build a CONFIGxx member

After dynamic changes have been made to a system it is recommended to update the corresponding CONFIGxx member to reflect these changes. HCD provides a function to build a CONFIGxx member containing the CHP, DEVICE, and SWITCH statements of the local system or of the selected system in a sysplex.

A CONFIGxx member can be built by:

- Selecting the **Build CONFIGxx member** action from the Activate or Verify Configuration panel (for the local system)
- Selecting the **Build CONFIGxx member** action from the Active Sysplex Member panel (for a system in a sysplex)
- Using a batch utility (see “Build I/O configuration data” on page 339 for details)

After selecting **Build CONFIGxx member**, the Identify System I/O Configuration panel is displayed (see Figure 100 on page 228). After selecting a system, and an I/O cluster name for managed channel paths, the Restrict Ports Eligible for Dynamic CHPID Management panel is displayed if the configuration contains managed channel paths for the selected I/O cluster. This panel shows all control units known by the selected system and manageable by DCM and their switch ports set to eligible for DCM (indicated by a 'Y'). You can specify ports as ineligible for DCM by overtyping 'Y' with 'N'.

Restrict Ports Eligible for Dynamic CHPID Management

Row 1 of 39

Command ==> _____ Scroll ==> CSR

Type 'N' to restrict ports related to managed CHPIDs from being used by dynamic CHPID management.

Processor ID: FR38LPAR Partition: F38H OS Configuration ID: B710
I/O Cluster name: UTCPLX38

SW.Port	----- Last digit of port address -----															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
65.4	-	-	N	-	-	-	Y	-	-	-	-	-	Y	Y	Y	-
65.B	-	-	Y	-	-	-	Y	-	-	-	-	-	Y	Y	Y	-
66.4	-	-	Y	-	-	-	Y	-	-	-	-	-	Y	Y	Y	-
66.B	-	-	Y	-	-	-	Y	-	-	-	-	-	Y	Y	Y	-
67.A	-	-	Y	-	-	-	Y	-	-	-	-	-	Y	Y	Y	-
67.B	-	-	Y	-	-	-	Y	-	-	-	-	-	Y	Y	Y	-
68.A	-	-	Y	-	-	-	Y	-	-	-	-	-	Y	Y	Y	-
68.B	-	-	Y	-	-	-	Y	-	-	-	-	-	Y	Y	Y	-
6D.C	-	-	Y	-	-	-	Y	-	-	-	-	-	-	-	-	-
6D.D	-	-	Y	-	-	-	Y	-	-	-	-	-	-	-	-	-
6E.C	-	-	Y	-	-	-	Y	-	-	-	-	-	-	-	-	-

Figure 111. Restrict Ports Eligible for Dynamic CHPID Management

The Build CONFIGxx Member panel is then displayed.

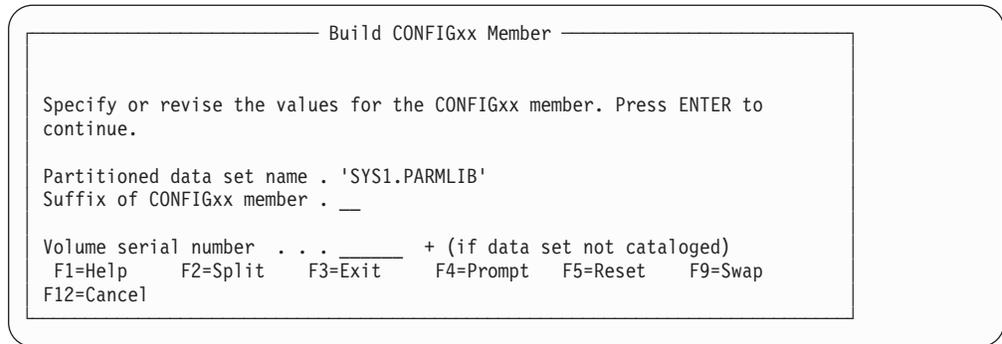


Figure 112. Build CONFIGxx Member

The initial value for the partitioned data set name is 'SYS1.PARMLIB'.

If the specified CONFIGxx member already exists, the Confirm Update CONFIGxx Member panel is displayed.

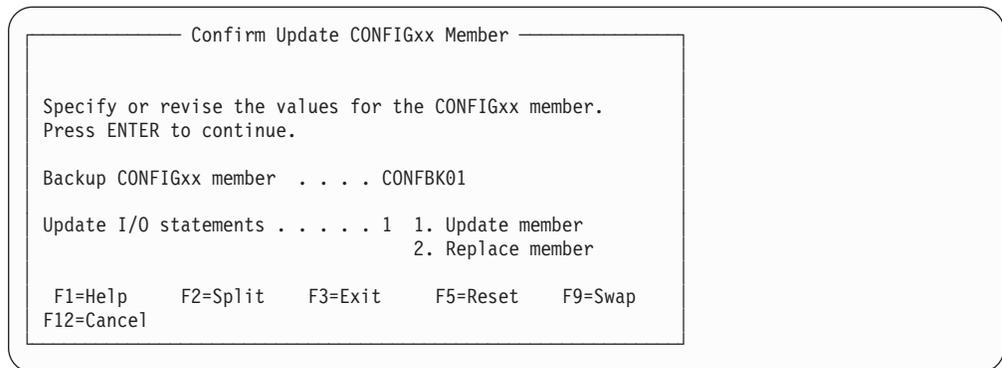


Figure 113. Confirm Update CONFIGxx Member

If you select Update member, the CHP, DEVICE, and SWITCH statements are replaced and all other statements remain unchanged. If you select Replace member, the content of the CONFIGxx member will be CHP, DEVICE, and SWITCH statements exclusively. All other statements formerly present in the member will be removed.

The following illustrates sample generated statements:

```
* CHP, DEV AND SWITCH STATEMENTS GENERATED BY
* BUILD CONFIGXX UPDATE REQUEST
* 2001-01-09 13:56:28 IODF: BOKA.IODF38
* PROCESSOR: FR38LPAR PARTITION: F38H OS CONFIGURATION ID: B710
* I/O CLUSTER: UTCPLX38
CHP (00,01,04),ONLINE
CHP (05),ONLINE,MANAGED
CHP (06,07,08,09,0A,0B,0C,0D,0E,10),ONLINE
CHP (11),ONLINE,MANAGED
.....
DEVICE (0B00-0B1F),(1C),ONLINE
DEVICE (1400-143F),(0C,22,33),ONLINE
DEVICE (1440-147F),(10,1C,44),ONLINE
.....
SWITCH (B565,42),NODCM
SWITCH (B565,B6,BC-BE),DCM
```

The default name for the backup member is CONFbkxx. If the name is blanked out, no backup is saved.

You can also invoke this task in batch mode. See “Build I/O configuration data” on page 339 for a description of the job control information that you need to specify when building a CONFIGxx member.

Process the Display M=CONFIG(xx) command

HCD provides a dialog function to compare the information in the CONFIGxx member for the system in a sysplex with the hardware configuration. The comparison is carried out at the target system and any responses are displayed in a message list.

You can invoke the function from the Active Sysplex Member List by selecting the action *Process DISPLAY M=CONFIG(xx) command*. This displays the Process Display M=CONFIG(xx) Command panel.

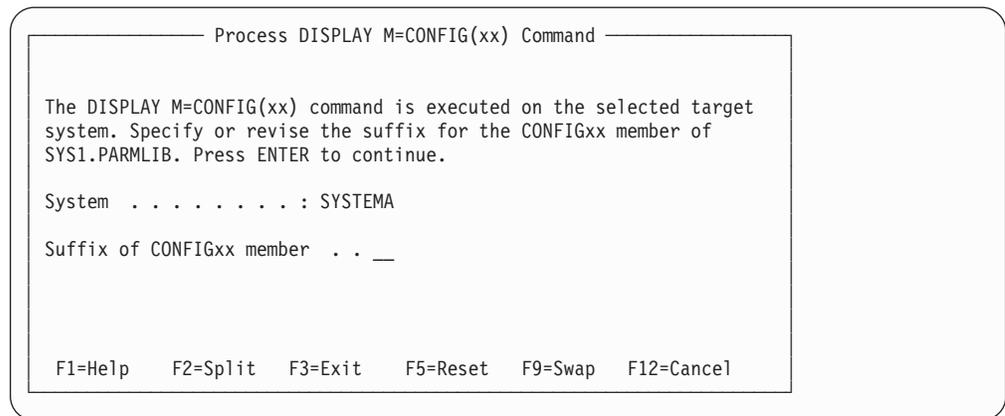


Figure 114. Process Display M=CONFIG(xx) Command

Here, the suffix for the member to be used must be specified.

The results of this action are displayed in the HCD message panel.

Switch IOCDS for next POR

HCD allows you to specify an IOCDS that will be used for the next POR either while building IOCDSs or as a separate action without the need to build an IOCDS. Depending on the environment you are working, you have to use different panels to switch the IOCDS:

- Switch IOCDS for processor without SNA address defined
- Switch IOCDS for a processor in a z Systems cluster with SNA address defined
- Switch IOCDS for systems in a sysplex

Switch IOCDS for processor without SNA address

The following procedure is only recommended for processors that do *not* have an SNA address defined. For a detailed description of the following dialog sequence, refer to “Build an IOCDS” on page 212.

1. On the *Primary Task Selection* panel, specify the name of a production IODF and select *Activate or process configuration data*.
2. From the resulting panel select *Build IOCDS*. HCD displays the Processor List.
3. On the Processor List, select the processor and press the Enter key. HCD displays the IOCDS List (see Figure 95 on page 212).

- On the IOCDS List, select the IOCDSs you want to use for next POR and select *Switch IOCDS* from the context menu (or action code **S**).

If the HSA token is available, the HSA token is compared with the processor token. If the HSA token matches the processor token, the *Switch IOCDS* action is performed. A warning message is issued, if the date in the IOCDS update record is an earlier date than the date of the last CSS update. If the HSA token does not match the processor token in the IODF, the action *Switch IOCDS* is not performed.

If the HSA token is not available, the serial number of the processor defined in the IODF is compared with the serial number of the active processor. If the serial numbers cannot be found, the types of the processors are compared. If the processor definition in the IODF matches the active processor, the action *Switch IOCDS* is performed, otherwise the procedure is not performed.

Switch IOCDS for processors in a z Systems cluster with SNA address defined

The following procedure describes how to build an IOCDS for processors in a z Systems cluster *with* an SNA address defined. For a detailed description of the following dialog sequence, refer to “Build z Systems cluster IOCDSs” on page 215.

- On the *Primary Task Selection* panel, select *Activate or process configuration data* and from the resulting panel select *Build and manage z Systems cluster IOCDSs, IPL attributes and dynamic I/O changes..* The *z Systems Cluster List* is displayed (see Figure 96 on page 215).
- On the *z Systems Cluster List*, select the CPCs for which you want to switch the IOCDSs and *Work with IOCDSs* from the context menu (or action code **S**). HCD displays the IOCDS List (see Figure 97 on page 216).
- Use the *Switch IOCDS* action (or action code **S**) to mark an IOCDS as the IOCDS that is used for next POR. The Status field will be set accordingly.

You can only switch to an IOCDS that has an IOCDS/HSA token match or to an IOCDS of a processor that is not activated ('POR-required' status).

Switch IOCDS for systems in a sysplex

In addition to dynamically activating a sysplex, you can also specify the IOCDSs to be used for the next POR.

- Select one or more systems from the Active Sysplex Member List (see Figure 102 on page 235) and the *Switch IOCDS for next POR* action from the context menu (or action code **S**). The Switch IOCDS panel is displayed.

Switch IOCDS

Row 1 of 3

Command ==> _____ Scroll ==> PAGE

Specify the IOCDS(es) for next POR, then press Enter.

System Name	Processor ID	Config. ID	EDT ID	Switch IOCDS	Active IODF
IRD4	ECL2	MVSVM	00	—	SYS4.IODF10
IRD5	ECL2	MVSVM	00	—	SYS4.IODF10
IRD6	R35	MVSVM	00	—	SYS4.IODF10

***** Bottom of data *****

- In the column Switch IOCDS, specify the IOCDS that is to be used for the next POR and press the Enter key.

| **Note:** If you prompt for IOCDS, you might see outdated IOCDS update dates, if it
| was not possible with the last write IOCDS action to update the IOCDS records in
| the IODF. To ensure update of the IOCDS dates, use option 2.2 (write IOCDS) for
| the processor in focus.

Specify an IODF for IPL

z/OS HCD Planning gives a detailed description of how to specify an I/O configuration at IPL. This topic summarizes the main aspects you have to consider when specifying an IODF for IPL.

Notes:

1. A production IODF must have a single extent. If the production IODF has multiple extents, the IPL process results in a WAIT state (wait state code '0B1', reason code '002'). HCD issues error message CBDA009I if a production IODF cannot be built in a single extent.
2. A production IODF must not be allocated with the SMS EXTENDED attribute. If the production IODF is allocated as an EXTENDED LINEAR dataset, the IPL process results in a WAIT state (wait state code 0B1, reason code 005).

IODF processing at IPL

When you perform an IPL, the production IODF that defines the configuration to the system is selected and used.

On the LOAD parameter, you specify the device containing the IODF for IPL and the identifier for the LOADxx member. In the LOADxx member of SYSn.IPLPARM or SYS1.PARMLIB, you identify the IODF by the IODF statement. The IODF statement consists of an IODF *prefix* and an IODF *suffix*.

- The IODF prefix is an 8-byte high-level qualifier of the IODF data set name. For example, BPAN is the IODF prefix for the IODF data set BPAN.IODF01.
- the IODF suffix is the two-digit hexadecimal number that is part of the IODF name. For example, 01 is the IODF suffix for IODF01. If you do not specify a suffix, the system searches for an IODF sequentially in a numerically ascending order starting with the IODF suffix 00. If you specify ** as the suffix, the system uses the descriptor fields to find the current IODF.

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

1. The system first searches the IODF volume for SYS0.IPLPARM through SYS9.IPLPARM, in that order. Therefore, it is recommended to use SYS0.IPLPARM for best IPL performance.
2. If it does not find a SYSn.IPLPARM, it searches the IODF volume for a SYS1.PARMLIB.
3. If it does not find SYS1.PARMLIB on the IODF device, it searches for SYS1.PARMLIB on the IPL device.
4. If it does not find a SYS1.PARMLIB on the IPL device, a coded non-restartable wait state is loaded (WAIT code X'0B1').

For a detailed description of this process refer to *z/OS HCD Planning*.

Chapter 10. How to print and compare configuration data

Overview

This information unit describes how to:

- Print configuration reports (channel subsystem, switch, OS configuration data, and CTC connections)
- Print a report of the I/O paths of the actual system compared to the defined I/O configuration
- Print a report of the supported hardware or an I/O definition reference
- Create or view a graphical report of the I/O configuration
- Compare functions (IODFs and CSS/operating system views)
- Print list panels
- View and print the activity log

Print configuration reports

You can use HCD to generate several types of reports about the configuration data in an IODF:

- Channel Subsystem (CSS) Report
- Switch Report
- Operating System (OS) Report
- CTC Connection Report
- I/O Path Report
- Supported Hardware Report
- I/O Definition Reference

In the HCD profile definition, you have the option of printing textual reports in upper case only or defining the number of lines per page (see “Options for text reports” on page 24).

Examples of these reports are shown in Appendix B, “Configuration reports,” on page 399.

Channel Subsystem Report

The Channel Subsystem Report contains all configuration data that is used by the channel subsystem. If the IODF contains data for more than one processor or logical partition, you can limit the report to the data for one processor or partition. If you limit the report to one partition, it will generate information only for channel paths, which have the partition in the access list. Channel paths that have that partition in a candidate list will not be taken into consideration.

You can select four types of reports:

- **CSS summary reports** include summary reports about:
 - Processors
 - Channel subsystems
 - Partitions

- PCIe functions
- PCHIDs
- IOCDs
- Channel paths
- Control units
- Devices

The processor and partition reports are not printed if you limit the CSS summary reports to the data for one processor or partition.

- **Channel path detail reports** include reports about:
 - Channel paths
 - CF channel path connectivity
- **Control unit detail report**
- **Device detail report**

Switch Report

The Switch Report contains details about the switch definition, its configurations and the port definitions.

If the IODF contains data for more than one switch, you can limit the report to the data for one switch and the configurations for this switch. In this case, you do not get a switch summary report.

Operating System Report

The Operating System Report contains the configuration data that is used by z/OS or z/VM. If the IODF contains data for more than one operating system, you can limit the report to the data for one operating system. You can select three types of reports:

1. The OS device report includes reports about operating systems and OS devices. The operating system summary report is not printed if you limit the OS device report to the data for one operating system.
2. OS console report
3. EDT report (MVS-type only)

CTC Connection Report

The CTC Connection Report contains CTC connections of your configuration that are defined through a switch or point-to-point. In case of incorrect definitions, the report also contains a list of messages with diagnostic information.

If the IODF contains more than one processor or logical partition, you can limit the report to the data for one processor or partition.

For capabilities and restrictions that apply to the presentation of CTC connections in the configuration diagram and the CTC Connection Report, refer to “Restrictions applying to the CTC Connection List” on page 157.

I/O Path Report

The I/O Path report shows the physically sensed I/O paths (with physical types) of the active system compared with the logical definitions of the paths (also the object types) of a specific IODF.

On the Limit Reports panel (Figure 116 on page 256) the active configuration to sense the configuration from, can be specified by indicating a SYSPLEX and/or SYSTEM name. If nothing is specified, the data is taken from the local system.

For more information, see “Job statement information used in panels” on page 72.

See “Prerequisites” on page 9 for the prerequisites for the I/O Path report.

Supported Hardware Report

The Supported Hardware Report contains information about the processors, control units, and devices supported in your installation. This report can only be generated using the batch facility as described in “Print configuration reports” on page 343.

This report is generated directly from the UIMs. Therefore, it reflects the latest UIM levels installed.

I/O Definition Reference

The I/O Definition Reference contains a description of the parameters to define the device to the Channel Subsystem, and a description of the parameters and features to define the device to the operating system.

This report is generated directly from the UIMs. Therefore, it reflects the latest UIM levels installed.

This report can only be generated using the batch facility as described in “Print configuration reports” on page 343.

How to print a textual report

1. On the *Primary Task Selection* panel, select *Print or compare configuration data*.
2. On the resulting Print and Compare Configuration Data panel, select *Print configuration reports*. HCD then displays the following panel:

```
Print Configuration Reports

Select the types of report you want, and specify the values below.

IODF name : 'DOCU.IODF01'

Types of report          Limit report(s)
- CSS report             1  1. Yes
- Switch report          2. No
- OS report
- CTC connection report
- I/O path report

Job statement information
//      JOB (ACCOUNT), 'NAME'
// *
// *
// *
// *
// *

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

Figure 115. Print Configuration Reports

3. Enter the required data.

When you select a CSS or OS report an additional panel appears on which you can select one or more report types.

If a data set is pre-allocated the logical record size must be 133. You can allocate the report output data set HCDRPT using the job step name GO.

4. When you select to limit the reports, possible for CSS, Switch, OS and CTC connection reports, the Limit Reports panel appears that allows you to specify a processor ID, partition name, OS configuration ID and a switch ID. When you select an I/O Path report, the Limit Reports panel always appears. This is because limiting an I/O Path report is required. Default values for the processor ID, the partition name (for an LPAR processor) and the OS configuration ID are then already filled in. These values are based on the active configuration. The system name identifies the system of a sysplex for that the I/O Path report is to be generated. The default is the local system. The sysplex name specifies the sysplex of the system for that the I/O Path report is to be generated. If you specify the sysplex, you must also specify the system name. If you do not specify the sysplex, the system name is the VTAM application name of the host that the I/O Path report is to be generated for. If you selected to print more than one report type, the limitations specified on the Limit Reports panel apply to all of them.

When limiting a CSS report to a single partition, the report will show channel paths, control units and devices attached by the access list as well as those attached by the candidate list.

Limit Reports

To limit the reports, specify the following criteria related to the IODF in access.

	Applicable for:
Processor ID _____	+ CSS, CTC, I/O path reports
Partition name . . . _____	+ CSS, CTC, I/O path report
OS configuration ID _____	+ OS, I/O path report
Switch ID _____	+ switch report

Specify the sysplex and system name to gather the actual configuration from. (Blanks default to the local system.)

Sysplex name _____	I/O path report
System name _____	I/O path report

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

Figure 116. Limit Reports

The submitted job only starts if the IODF is accessed in read mode. If it is accessed in update mode, the job waits until you access another IODF or exit HCD.

You can also print reports using the batch mode. See “Print configuration reports” on page 343 for a description of the job control information that you need to specify when printing a report.

Create or view graphical configuration reports

HCD offers you to print and view a graphical representation of the I/O configuration based on the definitions in the IODF. The reports can be either stored in a data set for printing on an AFP printer (such IBM 3820 or IBM 3800) or via GDDM later on, or displayed on an IBM 3270 terminal with graphical capability.

The graphical report function allows you to print or view five types of reports:

- The **LCU report** shows all logical control units defined for one processor.
- The **CU report** takes a control unit as focal point and shows the connections to the processors and the devices of the IODF. On request, it shows the switches as well.
- The **CHPID report** shows the defined channel paths for a processor and the switches, control units, and devices attached to the CHPID.
- The **Switch report** takes a switch (ESCON director) as focal point and shows the processors, chained switches, and control units with devices attached to the switch.
- The **CF connection report** takes a coupling facility as focal point and shows all connections that exist between the coupling facility and the other processors defined in the IODF.

Prerequisites

For printing

To process the reports for printing you need one of the following:

- BookMaster Release 3.0 or higher
- DCF/GML Release 4.0
- GDDM Version 2.1 or later

To print the reports you need an AFP printer, such as IBM 3820 or IBM 3800 (not required for GDDM).

To store the output in GDF format, you have to use a terminal with a screen size of 80 columns, for example a 3278-2.

In the HCD profile, specify whether the output of this function can be processed with BookMaster, DCF, GML, or GDF (keyword `GCR_FORMAT`). BookMaster is the default. To use DCF or GML format, specify a mono- space font using the keyword `GCR_FONT`. For example, specify `GCR_FONT = X0GT20` (Gothic Text 20-pitch) for a 3820 printer. For more information about the HCD profile, refer to “Defining an HCD profile” on page 19.

For viewing

To view the report on an IBM 3270 terminal with graphical capability, GDDM must be installed on your system. Refer to “Setting up HCD” on page 15 on how to setup the GDDM support.

Use a terminal with a screen size of 80 columns, for example a 3278-2. This display function does not work on terminals (or terminal emulations) with a screen size of 132 columns. HCD uses ISPF to create the GDDM display, which means that terminals running in partition mode or terminals with multiple screen widths, including 3290 and the 3278 Mod 5, are not supported for graphics interface mode.

Rotate printout

Specify a parameter to print the report in landscape format, that means to rotate the printout by 90 degree.

2. Print the report.

HCD tries to display a report on one page. If a report is too large for one page, HCD divides the report into segments and shows each segment on an extra page. If a CU report, for example, shows more than 8 control units, HCD shows the control units of the same type in one segment on an extra page. If a segment is too large for one page, HCD continues the segment on the next page. You can specify the GCR_COMPACT=YES keyword in the HCD profile to see more objects on one page.

For an example of a report, see “Graphical configuration reports” on page 454.

Viewing the output

The following figure shows an example of the panel when viewing a report.

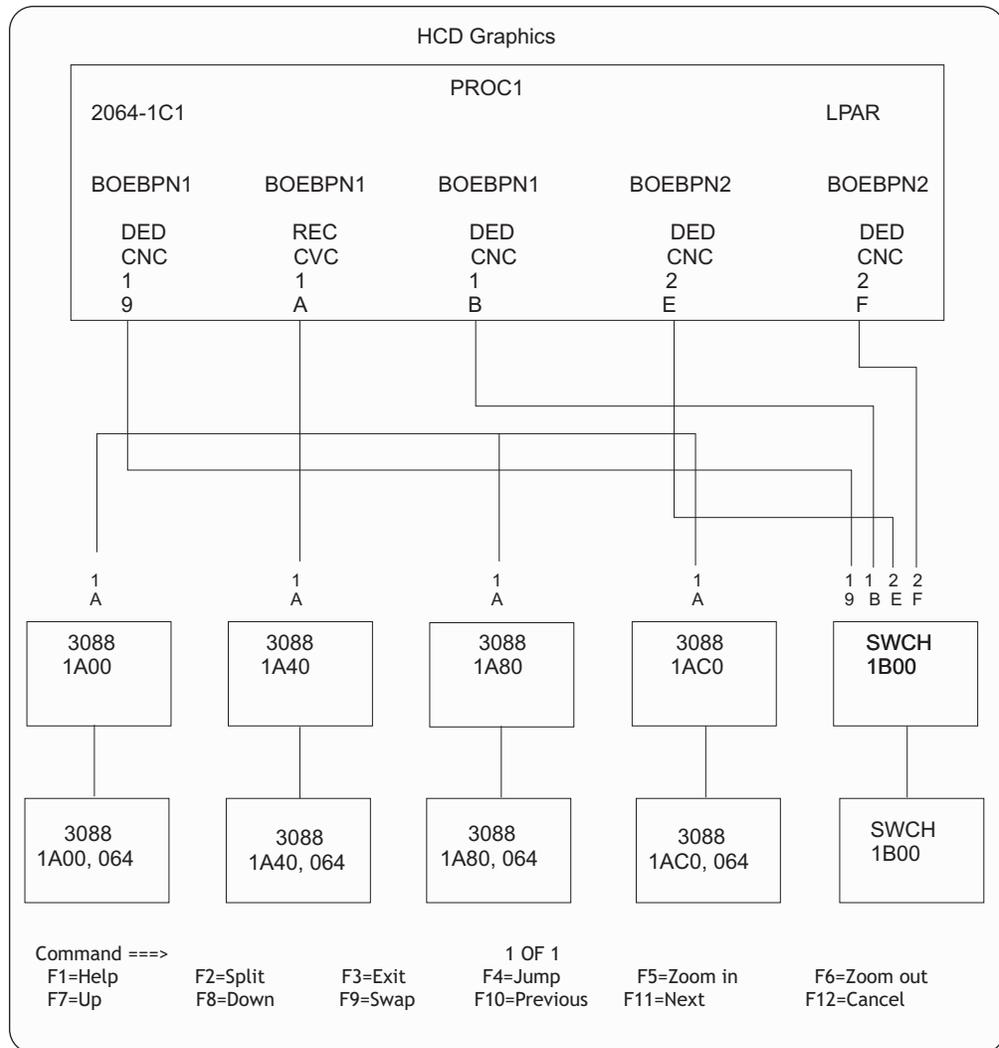


Figure 117. Viewing a sample report

HCD tries to display the entire configuration on one panel. If a report is too large for one panel, HCD divides the report into segments and shows each segment on an extra panel. If a CU report, for example, shows more than 8 control units, HCD

shows the control units of the same type in one segment on an extra panel. You can move from segment to segment by using the Next and Previous function keys.

If a segment of control units is too large for the panel, you can scroll upwards, downwards, to the left and to the right.

The following list describes specific function keys while viewing a graphical report.

F4=Jump

You can use this function only when you have displayed the graphical report using the task *Create or view graphical configuration report* on the *Primary Task Selection* panel. It displays the action list of the HCD dialog that contains the object you selected with the cursor. Any change made to the configuration on the action list will not be reflected in the graphical report when you return to it. Use the REFRESH command to reflect the changes made to the objects currently shown on the display (see “Refresh Command” on page 263).

F5=Zoom in

Makes the graphical display bigger, so you can see the details of an object. The position of the cursor identifies the lower left corner of the part you want to enlarge. If the cursor is not positioned, HCD takes the center of the currently displayed report.

F6=Zoom out

Makes the graphical display smaller, so you can see more of a report on one panel.

F7=Up

Scrolls upwards.

F8=Down

Scrolls downwards.

F10=Previous

Moves to the previous segment of a report, if any.

F11=Next

Moves to the next segment of a report, if any.

F19=Left

Scrolls to the left.

F20=Right

Scrolls to the right.

SAVE command

You can use the SAVE command to store a graphical configuration displayed on your screen in a member of a partitioned data set (PDS) in GDF format for printing with GDDM.

Before you use this command, the partitioned data set has to be allocated to ddname ADMGDF. The records of this PDS must have a record length of 400.

Specify SAVE (or just SA) and the member name, into which you want to save the data, on the command line. The syntax of the SAVE command is as follows:

SAVE command syntax



member_name

specifies the name of the member to contain the graphical configuration you want to print. Use a different name for each SAVE command, otherwise the data will be overwritten. The name may be up to eight characters long.

LOCATE command

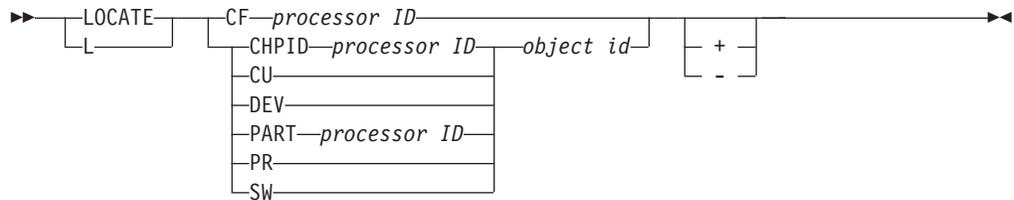
You can use the LOCATE command to center and highlight a specific object of the report. When you locate a partition, channel paths, or coupling facility partition, the connections to the object are highlighted instead of the object itself.

Specify LOCATE (or just L) and the object you want to locate in the command line. For objects that are associated with another object (for example, CHPIDs that are associated with a processor), you have to specify this object as well.

Locating multiple objects with the same ID: The LOCATE command first searches for an object in the currently displayed panel. If the object is not found it starts with the first segment and continues to the right until an object has been found. To find the other objects with the same ID, use the + and - parameter of the LOCATE command. The + parameter searches for the next object to the right, the - parameter to the left.

The syntax of the LOCATE command is as follows:

LOCATE command syntax



CF Coupling Facility

CHPID
Channel path

CU Control unit

DEV Device

PART Partition

PR Processor

SW Switch

Example: To locate CHPID 27 of processor SYSA, type
L CHPID SYSA 27

| **Locate processing on device lists:** Locate searches the leftmost column of the
| active panel's list for a specified object identifier.

| If the specified object exists within the list, it is displayed at the top row of the list
| and the cursor locates it at this object.

| If a non-existing object within the range of the list is specified, an existing object
| whose character string precedes the specified one is displayed in the top row of
| the list and the cursor locates it at the next object.

| If a non-existing object outside the range of the list is specified, an error message is
| reported.

| **Note:** For the grouped I/O device list, the locate applies only to the device
| number without considering the range. If the specified device number exists within
| the range of a device group, the locate applies as if searching for a non-existing
| object within the list (for example: it displays the device group at the top row and
| places the cursor on the next object). In order to locate a specific device within the
| range of a device group, you need to expand the list by entering 's' on any device
| and then issue the locate on the expand device.

Refresh Command

When you jump to an action list using F4=Jump and change objects in this list, use the REFRESH command on return to the graphical display to refresh the graphic with the changes made.

REFRESH applies to all objects that were currently shown on the graphical report when pressing the F4=Jump key, that is, added objects or those, for which you changed the ID, will not appear in the refreshed graphic.

Valid abbreviation of the REFRESH command is RE.

How to print list panels

You can use the SAVE command to save the data that is currently displayed on HCD list panels into a data set. The data set can be used for printing.

You can also save and print lists that are filtered. If you use the *Set Filter* option from the *Filter* action bar choice, you can, for example, print all channels of a processor that are not connected to a control unit.

The SAVE command is available on the following lists:

- Operating system configuration List
 - EDT List
 - Esoteric List
 - Esoteric/Device List
 - Generic
 - Generic/Device List
 - Console List
- Switch List
 - Port List
 - Switch Configuration List
 - Port Matrix
- Processor List
 - PCIe Function List
 - Channel Subsystem List
 - Channel Path List
 - Partition List

- Control Unit List
- I/O Device List
- Configuration Package List
 - Configuration Package Object List
- CTC Connection List
- I/O Path List
- I/O Autoconfiguration lists:
 - Discovered New or Changed Controller List
 - Proposed Control Unit List
 - Proposed Control Unit / Device List
- CPC Image List

On these list panels, perform the following steps:

1. Enter the command

SAVE

2. The Save List appears:

Save List

Specify the following values.

Output data set _____

Additional remarks (for example, the filter criteria)
with DCM **2** _____

The output data set can be a sequential data set or a member of a partitioned data set. If the data set does not exist, it will be automatically allocated (record length 300, record format fixed block). The name of the data set is saved for the next HCD session.

In addition you can specify two lines of optional comments that appear under the header of your output.

3. A result of a printed data set may look like the following example. Note that the column headers are the same as shown on the panel itself.

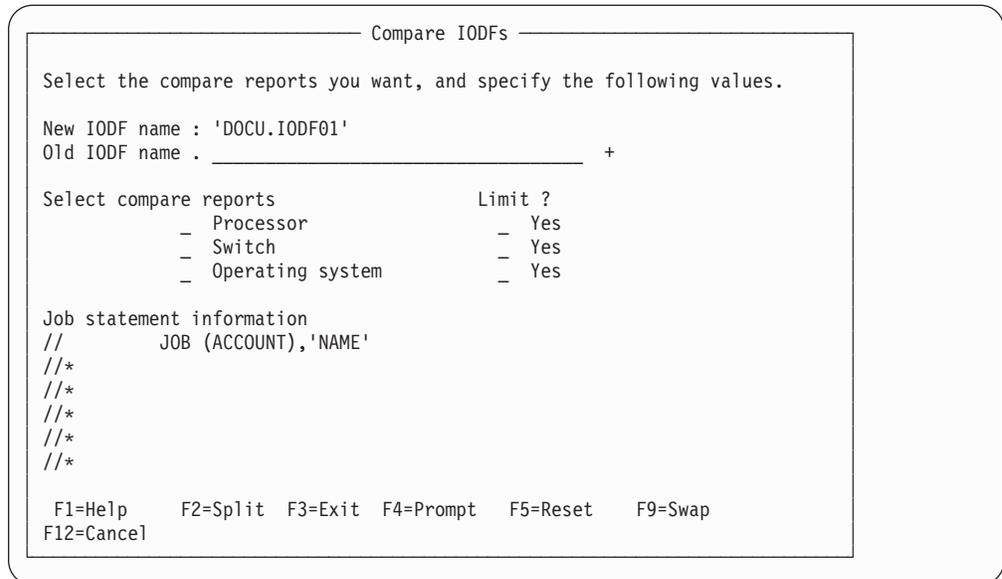


Figure 119. Compare IODFs

On this panel, select one or more compare report(s). In addition, you can set the limit option. When the limit option is set, the related limiting panels will come up.

On the *Limit Processor Compare Reports* panel, you can limit the processor compare reports by selecting one or more of the specific compare reports. You can limit the reports by specifying values for a processor or either by specifying values for a channel subsystem or a partition. It is possible to compare an SMP processor to a channel subsystem of an XMP processor. If you want to limit by processor, you must specify the processor IDs for both IODFs. If you limit the processor compare report by partition name, you receive the following results:

- The report will contain the channel subsystem in which the partition is defined.
- Channel path compare will only contain channel paths which have the limiting partition in their access or candidate list.
- Control unit compare will only include the control units related to channel paths which have the limiting partition in their access or candidate list.
- Device compare will only include the devices connected via channel paths which have the limiting partition in their access or candidate list.

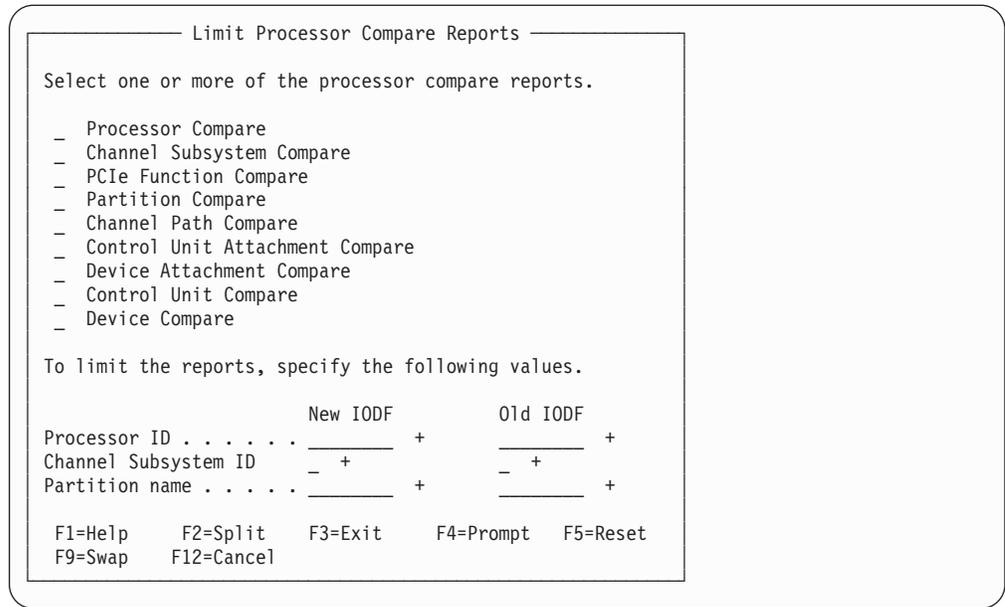


Figure 120. Limit Processor Compare Reports

On the *Limit Switch Compare Reports* panel, you can limit the switch compare reports by one or more of the specific compare reports. In addition, you can limit the reports by specifying a switch ID for both, the new and the old IODF.

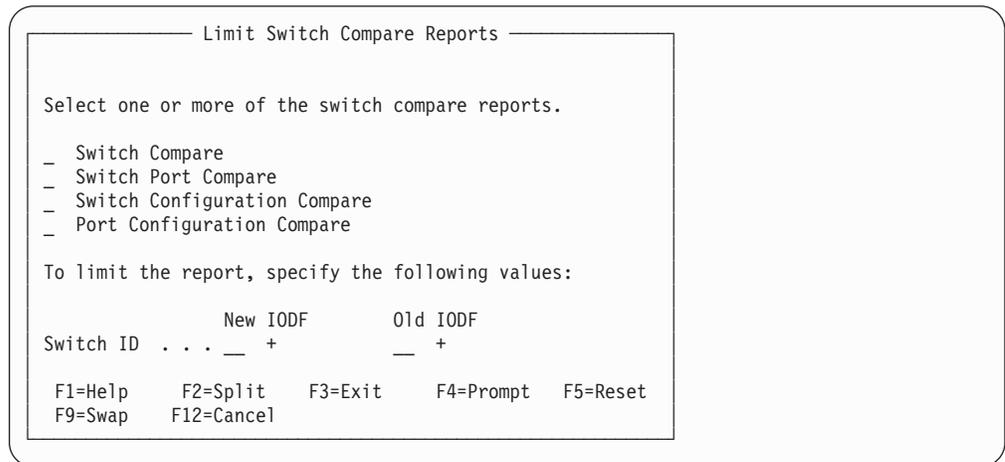


Figure 121. Limit Switch Compare Reports

On the *Limit Operating System Compare Reports* panel, you can limit the operating system compare reports by one or more of the specific compare reports. In addition, you can limit the reports by specifying an operating system ID for both, the new and the old IODF.

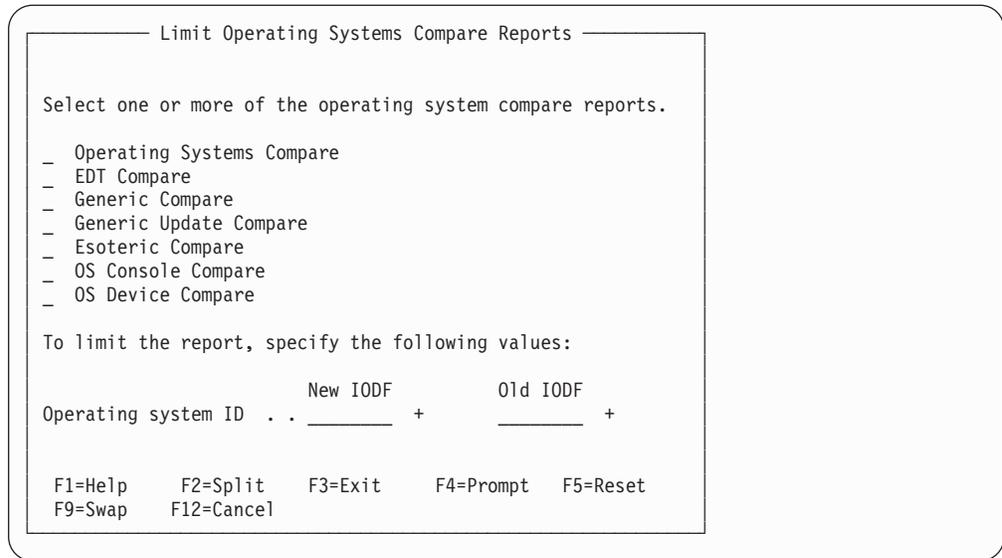


Figure 122. Limit Operating System Compare Reports

How to print a Compare IODFs Report

After you have selected the specific pairs for the compare reports, you can decide what print options to use. The print options are shown on the Select Print Options panel, see Figure 123.

If you do not select an option (by only pressing the Enter key), the default print options are used (Print inserted data, and Print deleted data).

Examples of these reports are shown in Appendix B, “Configuration reports,” on page 399.

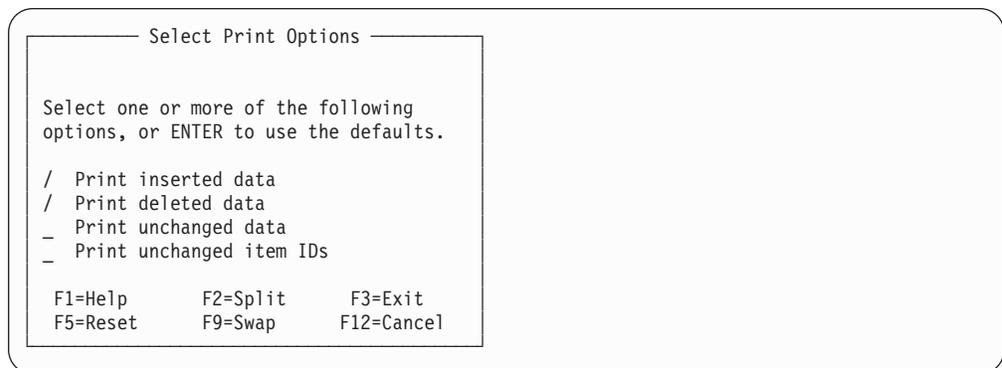


Figure 123. Select Print Options (for Compare IODFs only)

When you do not select any option, HCD prints a report of

- IDs of added or deleted objects, and those objects, that have added or deleted relations
- IDs of added relations
- Attributes of objects that are different in both IODFs

When you select the Print inserted data option, HCD prints a report of all attributes, and relations of added objects.

When you select the Print deleted data option, HCD prints a report of all attributes, and relations of deleted objects.

When you select the Print unchanged data option, HCD prints a report of all attributes, and relations of unchanged objects.

When you select the Print unchanged item IDs option, HCD prints a report of the IDs of unchanged objects. But this applies only, if the Print unchanged data option is not selected.

Compare CSS / operating system views

You can use the *Compare CSS / operating system views* function to 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. By using this function you can find out the differences between a hardware (channel subsystem/CSS) and software (operating system/OS) definition in the currently accessed IODF.

```
Compare CSS / OS Views

Specify the following values.

IODF name . . . : 'DOCU.IODF00.FP.NEW'

CSS view
Processor ID . . . . _____ + Partition name . . _____ +

OS view
Configuration ID . . _____ + 2  1. Print all data
                                     2. Print different data

Job statement information
//      JOB (ACCOUNT),'NAME'
//*
//*
//*
//*
//*

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

Figure 124. Select Print Options (for CSS/OS Compare only)

When using the *Compare CSS / operating system views* function, you are offered the possibility of selecting what to print on the *Compare CSS / OS Views* panel, shown in Figure 124. The Print all data option prints a report of all devices either defined in the CSS or the OS.

The Print different data option prints a report of the devices that differ as follows:

- Defined for the CSS, but not for the OS.
- Defined for the OS, but not for the CSS.
- Defined for both, but of different device type.

If you limit the Compare CSS/OS Views Report for the CSS-side to one partition, it will generate information only for those devices that are attached to the channel paths, that have the limiting partition in the access or candidate list.

View and print the HCD activity log

The HCD activity log is described in “Activity logging and change logging” on page 50.

To look at the activity log for an IODF,

1. Select *Print or compare configuration data* from the *Primary Task Selection* panel.
2. On the Print or Compare Configuration Data panel, select *View the activity log* or *Print the activity log*.

The log has the same format in both cases. For browsing, the log is displayed by the ISPF/PDF browse facility. For printing, the log is written to the ISPF list data set.

Note: HCD maintains an activity log only if this was requested when the IODF data set was created.

Chapter 11. How to query supported hardware and installed UIMs

Overview

This information unit describes how to view system data about:

- Supported processors
- Supported switches
- Supported control units
- Supported devices
- Supported installed UIMs

Your z/OS system has several tables and modules that contain data about the general characteristics of processors, switches, control units, and devices in the system.

HCD uses this data to validate your configuration definition. You might want to look at it during the definition task; it helps you select the correct characteristics when you define your hardware units. You can also use the supported hardware report as a help when defining your configuration. See “Supported hardware report” on page 425 for an example of a supported hardware report and “I/O Definition Reference” on page 451 for an example of an I/O definition reference.

You can view the system data by selecting *Query supported hardware and installed UIMs* from the *Primary Task Selection* panel or the *Query* action from the action bar. Then select the subtask or pull-down choice that you want.

Query supported processors

The option List supported processors shows which processors are supported by the system. You can also see which features each processor supports, and its capabilities (such as what types of channel paths the processor supports).

```
*----- Supported Processors -----*
                                     Row 356 of 681 More:  >
Command ==> _____ Scroll ==> CSR

Select one to view more details.

Processor
Type-Model Support Level
2964-L96   2964 LinuxONE Emperor support
2964-L96   2964 LinuxONE Emperor support, UID
2964-NC9   2964 support, ISM, RCE, UID
2964-NC9   2964 support
2964-NC9   2964 support, ISM, RCE
2964-NE1   2964 support, ISM, RCE, UID
2964-NE1   2964 support
2964-NE1   2964 support, ISM, RCE
2964-N30   2964 support, ISM, RCE, UID
*-----*
```

Figure 125. Query supported processors

The two lines in Figure 125 on page 271 marked with **1** and **2** illustrate that processors with different support levels generate more entries in the list of supported processors.

Horizontal scrolling displays additional information on the processors.

A # preceding a line indicates that this line and the previous line belong together, because the information of a horizontally scrolled screen does not fit in one line.

```
*----- Supported Processors -----*
                                     Row 356 of 681 More: < >
Command ==> _____ Scroll ==> CSR

Select one to view more details.

Processor Supported
Type-Model Channel Path Types
2964-L96 FC,FTV,OSD,OSE,IQD,FCP,STV,OSC,OSN,OSX,OSM
2964-L96 FC,FTV,OSD,OSE,IQD,FCP,STV,OSC,OSN,OSX,OSM
2964-NC9 FC,FTV,OSD,OSE,ICP,IQD,FCP,STV,OSC,OSN,CIB,OSX,OSM,CS5
2964-NC9 FC,FTV,OSD,OSE,ICP,IQD,FCP,STV,OSC,OSN,CIB,OSX,OSM,CS5
2964-NC9 FC,FTV,OSD,OSE,ICP,IQD,FCP,STV,OSC,OSN,CIB,OSX,OSM,CS5
2964-NE1 FC,FTV,OSD,OSE,ICP,IQD,FCP,STV,OSC,OSN,CIB,OSX,OSM,CS5
2964-NE1 FC,FTV,OSD,OSE,ICP,IQD,FCP,STV,OSC,OSN,CIB,OSX,OSM,CS5
2964-NE1 FC,FTV,OSD,OSE,ICP,IQD,FCP,STV,OSC,OSN,CIB,OSX,OSM,CS5
2964-N30 FC,FTV,OSD,OSE,ICP,IQD,FCP,STV,OSC,OSN,CIB,OSX,OSM,CS5
*-----*
```

```
*----- Supported Processors -----*
                                     Row 356 of 681 More: < >
Command ==> _____ Scroll ==> CSR

Select one to view more details.

Processor Support Supported PCIe
Type-Model Level ID Protocols WI RI DP Fct
2964-L96 H160310 D,S,S4 Yes Yes Yes Yes
2964-L96 H160930 D,S,S4 Yes Yes Yes Yes
2964-NC9 H160930 D,S,S4 Yes Yes Yes Yes
2964-NC9 H150111 D,S,S4 Yes Yes Yes Yes
2964-NC9 H160310 D,S,S4 Yes Yes Yes Yes
2964-NE1 H160930 D,S,S4 Yes Yes Yes Yes
2964-NE1 H150111 D,S,S4 Yes Yes Yes Yes
2964-NE1 H160310 D,S,S4 Yes Yes Yes Yes
2964-N30 H160930 D,S,S4 Yes Yes Yes Yes
*-----*
```

```
*----- Supported Processors -----*
                                     Row 356 of 681 More: < >
Command ==> _____ Scroll ==> CSR

Select one to view more details.

Processor --Highest Number for-- -----Maximum Number of-----
Type-Model CUs Devices CHPIDs CUs LCUs Partitions CSS SCHS
2964-L96 FFFE FFFF 1536 8192 4096 90 6 4
2964-L96 FFFE FFFF 1536 8192 4096 90 6 4
2964-NC9 FFFE FFFF 1536 8192 4096 90 6 4
2964-NC9 FFFE FFFF 1024 8192 4096 90 6 4
2964-NC9 FFFE FFFF 1536 8192 4096 90 6 4
2964-NE1 FFFE FFFF 1536 8192 4096 90 6 4
2964-NE1 FFFE FFFF 1024 8192 4096 90 6 4
2964-NE1 FFFE FFFF 1536 8192 4096 90 6 4
2964-N30 FFFE FFFF 1536 8192 4096 90 6 4
*-----*
```

You can view channel path, control unit, and device information by placing the cursor in front of a processor and pressing the Enter key. As a sample of this type of information, you see the channel path information in the following diagram:

```

View Channel Path Information
-----
Row 1 of 17
Command ==> _____

ENTER to view information about allowed channel path type mixes.
Processor Type-Model . . . . . : 2964-N30
XMP, 2964 support

Supported Maximum --Max. Number per CHPID-- ---Supported--- Shared
Type Number UA Ranges Links Devices Time-out STADET
CIB 256 256 256 256 No No Yes
CS5 128 256 256 256 No No Yes
FC 320 256 256 32768 No Yes Yes
FCP 320 1 480 No No Yes
ICP 32 256 256 256 No No Yes
IQD 32 64 12288 No No Yes
OSC 48 1 254 No No Yes
OSD 48 16 480 No No Yes
OSE 48 1 255 No No Yes
OSM 48 16 480 No No Yes
F1=Help F2=Split F3=Exit F7=Backward F8=Forward
F9=Swap F12=Cancel F22=Command

```

Pressing the Enter key gives you a list of allowed channel path type mixtures. Pressing the Enter key again gives you similar information for control units and devices.

Note:

For System z processors, you can retrieve an explanation of the processor support level: Position the cursor on the processor support level description **1** and press PF1 to get an enumeration of functions provided by this support level.

Query supported switches

The option List supported switches shows the characteristics of each type of switch in the system, such as the port range and the supported channel attachments of each switch.

```

Supported Switches

ENTER to continue.

Switch Type Port Numbers Supported
Low - High Channel Attachments
FCS 00 - FF FC
2032 00 - FF FC
9032 C0 - FB CNC,CTC,CVC,CBY
9032-3 80 - FB CNC,CTC,CVC,CBY
9032-5 04 - FB CNC,CTC,CVC,CBY,FCV
9033 C0 - CF CNC,CTC,CVC,CBY

```

Query supported control units

The option List supported control units displays a panel showing a list of available control unit groups, for example the DASD control unit group. Select one control unit group to limit the list of supported control unit types. The Supported Control Units panel appears showing the characteristics of control unit types contained in a group.

```

----- Supported Control Units -----
Row 2 of 28 More: >
Command ==> _____ Scroll ==> CSR

Select a control unit to view the list of device types that can be
attached to the control unit.

-----Default for-----
Control Unit Supported Protocol I/O Concurrency
Type          Protocols Used      Level
2105          S,S4      S          2
2835-2        D          D          2
3380-CJ2      S          S          2
3830-2        D          D          2
3830-3        D          D          2
3851          D          D          2
3880-1        D,S        S          2
3880-11       D,S        S          2
3880-13       D,S        S          2
3880-2        D,S        S          2
3880-21       D,S,S4     S          2
3880-23       D,S,S4     S          2
3880-3        D,S        S          2

```

Horizontal scrolling displays additional information on supported channel path type attachments.

```

----- Supported Control Units -----
Row 2 of 28 More: <
Command ==> _____ Scroll ==> CSR

Select a control unit to view the list of device types that can be
attached to the control unit.

Control Unit Supported
Type          Channel Attachments
2105          CNC,FC,FCV
2835-2        BL,CVC,EIO
3380-CJ2      BL,CVC,EIO
3830-2        BL,CVC,EIO
3830-3        BL,CVC,EIO
3851          BL,BY,CVC,CBY,EIO
3880-1        BL,CVC,EIO
3880-11       BL,CVC,EIO
3880-13       BL,CVC,EIO
3880-2        BL,CVC,EIO
3880-21       BL,CVC,EIO
3880-23       BL,CVC,EIO
3880-3        BL,CVC,EIO

```

You can view which devices can be attached to a certain control unit type by placing the cursor in front of a control unit and pressing the Enter key. As a sample of this type of information, you see the CU - Device Attachment List:

```

----- CU - Device Attachment List -----
                                         Row 1 of 2
Command ==> _____ Scroll ==> PAGE

Following device(s) can be attached to control unit type
3990

ENTER to continue.

Device Type
3380
3390
***** Bottom of data *****

```

Query supported devices

The option List supported devices displays a panel showing a list of available device groups, for example the DASD device group. Select one device group to view characteristics of device types contained in this group. You can limit the list of device types to view only the device types supported by a specific operating system type. After selecting a group of devices, the Supported Device Type List appears.

```

----- Supported Device Type List -----
                                         Row 13 of 27
Command ==> _____ Scroll ==> PAGE

Select a device type to view the list of control unit type(s) to which the
device can be attached.

Device          --Support--      -Recommended- -Console-
Type           MVS  VM      Dyn.  Four- Multi-  Time-  STADET  NIP   VM
                Yes  Yes  Yes  Yes  No   Out   Yes  Elig.  Us
3370            Yes  Yes  Yes  No   Yes  Yes  No    No
3375            Yes  Yes  Yes  No   Yes  Yes  No    No
3380            Yes  Yes  Yes  No   Yes  Yes  No    No
3380-CJ2       Yes  Yes  Yes  No   Yes  Yes  No    No
3390            Yes  Yes  Yes  No   Yes  Yes  No    No

```

You can also see what control units each I/O device type can be attached to by placing the cursor in front of a device and pressing the Enter key.

As a sample of this type of information, you can see the Device - CU Attachment List:

```

Device - CU Attachment List
Row 1 of 11
Command ==> _____ Scroll ==> PAGE

Following control unit type(s) support the attachment of
device type
3380

ENTER to continue.

Control Unit Type
NOCHECK
3380-CJ2
3880-13
3880-2
3880-23
3880-3

```

Query installed UIMs

The option List installed UIMs shows which UIMs are available in the system and which I/O device types are supported by each UIM.

```

Installed UIMs
Row 1 of 51
Command ==> _____ Scroll ==> PAGE

Select one for a list of the devices it supports.

UIM Name  E Description
CBDUS001  UIM for 3330, 3333, 3340, 3344, 3350
CBDUS002  UIM FOR 3375, 3380 ,3390, 3995-151/153, 9345
CBDUS003  UIM for 3350P and 3351P
CBDUS004  UIM for 327x devices
CBDUS005  UIM for Magnetic Tape Devices
CBDUS011  UIM for 3800
CBDUS012  UIM for Unit Record devices
CBDUS013  UIM for 2305-2
CBDUS014  UIM for CTC Devices
CBDUS022  UIM for AFP1 and 3820 Printers
CBDUS023  UIM for 37xx and 7770
CBDUS024  UIM for 1030, 1050, 1050X, 115A, 2740, 2740C, 2740X, 376
CBDUS025  UIM for 2741P, 2741C, 83B3, TWX, WTTA
CBDUS026  UIM for BSC1, BSC2, BSC3
F1=Help   F2=Split   F3=Exit    F7=Backward F8=Forward
F9=Swap   F12=Cancel F22=Command

```

A Y (for Yes) in the E (for Error) column indicates that the respective UIM is in error and treated as not existing.

A # sign in front of a UIM name indicates that it cannot be selected because it is flagged in error.

For each of the installed UIMs you can view a list of supported devices by placing the cursor in front of a UIM and pressing the Enter key. As a sample of this type of information, you can see the View Supported Devices list:

View Supported Devices

Row 1 of 7

Command ==> _____ Scroll ==> PAGE

This UIM supports the listed device types.

UIM name : CBDUS002
UIM FOR 3375, 3380 ,3390, 3995-151/153, 9345

ENTER to continue.

Device Type	Generic -- or -- VM D/T	Description		
3375	3375	Direct Access Storage Device		
3380	3380	Direct Access Storage Device		
3380-CJ2	3380	Direct Access Storage Device		
3390	3390	Direct Access Storage Device		
3995-151	3390	Direct Access Storage Device		
3995-153	3390	Direct Access Storage Device		
9345	9345	Direct Access Storage Device		
F1=Help	F2=Split	F3=Exit	F7=Backward	F8=Forward
F9=Swap	F12=Cancel	F22=Command		

Chapter 12. How to migrate existing input data sets

Overview

This information unit describes how to:

- Prepare the input data sets before migrating them
- Migrate the input data sets using the HCD dialog
- Migrate the input data sets using the HCD batch utilities
- Replace existing configuration data through migration
- Change I/O configurations by editing data sets
- Understand and resolve errors that occurred during migration

HCD allows you to migrate existing configuration data that was defined in IOCP, MVSCP, and HCPRIO input data sets to an IODF.

You can also use the migration to create I/O definitions by editing control statements. Data sets containing the statements corresponding to a specific IODF can be generated using a batch utility. Refer to “Build I/O configuration data” on page 339 for details on which data sets can be built and how to run the build process.

When migrating from input data sets, HCD checks the syntax of the input statements and runs a validation process that checks that the definitions being migrated do not conflict with the I/O configuration rules and with existing definitions in the IODF or with other definitions being migrated.

If HCD detects an error in the input data sets, it issues messages after the migration process has ended.

Note: The migration function has a prerequisite to the High Level Assembler.

Migration sequence

If you want to migrate more than one input data set into a single IODF, comply to the following migration sequence:

1. Migrate all IOCP input data sets
2. Migrate all MVSCP or HCPRIO input data sets

LPAR considerations

If you have a combined IOCP/MVSCP input data set containing definitions for more than one LPAR and the same device number specified for more than one LPAR, migrate this input data set in the following way:

1. Migrate the input data set as IOCP only input data set.
2. Remove the duplicate device number definition and repeat the migration as an MVSCP only input data set. In the appropriate IODEVICE statement, specify the control unit number the device attaches to by means of the CUNUMBR parameter.

Preparing your input data sets for migration

Before you can successfully migrate IOCP, MVSCP, or HCPRIO input data sets, you may need to change the input because HCD does a more rigorous checking of the input statements than IOCP or MVSCP.

To ensure that the migration is successful and that the resulting IODF accurately reflects the physical configuration, ensure that your input data sets apply to the validation rules that will be described. For assistance when checking the definitions in the input statements, you can use the following possibilities:

- Select *Query supported hardware and installed UIMs* from the *Primary Task Selection* panel
- Use the *Query* action bar choice
- Use the batch facility "Print a Configuration or Supported Hardware Report"

Data requiring attention

This section details which data may need to be changed and how to change it to ensure a successful migration.

Control unit types

HCD checks whether a specified control unit type is valid. Review your input for invalid control unit types. If applicable, correct the UNIT parameter of the CNTLUNIT statement.

If you do not want to change the type in your input data set, you can edit the HCD profile and specify how a control unit type in the IOCP input data set is mapped to a control unit type in the IODF. Specify one or more of the following keyword:

```
MAP_CUTYPE = xxxxx,yyy-yy
```

```
xxxxx
```

is the control unit type specified in the IOCP input data set

```
yyy-yy
```

is the control unit type and model (optional) to be used in the IODF

For more information about the HCD profile, refer to "Defining an HCD profile" on page 19.

Control unit models

HCD requires, for certain control unit types, the specification of a model. For example, the IBM 3880 control unit requires a model specification. If a control unit type requires a model specification, and if you do not specify one in the input data set, HCD assigns a model to the control unit definitions based on the attached devices and the used control unit protocol. This control unit model is indicated as default model in the UIM (information message CBDA534I is issued). As processing goes on, it can be necessary to change the default model to another model to support the specified protocol (warning message CBDA536I is issued). Or, the default model is changed to attach a device type which is not supported by the default control unit model (warning message CBDA265I is issued).

Note: The sequence of messages is shown in reverse order in the migration log file since the messages are sorted according to decreasing severities. For an example, please see "Errors detected during assembly process" on page 322.

To assign a model to a control unit, change the UNIT parameter of the CNTLUNIT statement in the input data set. Append the model number separated by a dash to the control unit type specification. For example:

```
CNTLUNIT ...,UNIT=3880-23
```

To avoid changing the input data sets, you can also add a model number by using the MAP_CUTYPE parameter in the HCD profile as described under "Control unit types" on page 280.

Protocol support for control units

HCD checks the protocols supported by a control unit type. For example, in the IOCP input you may have an IBM 3745 with protocol S incorrectly specified as control unit type 3705. The IOCP program does not check the protocol S specification. To be accepted by HCD as valid input, you have to change the control unit type to 3745.

Device types

For IOCP input data sets: HCD checks the device types for validity and that they can be attached to the specified control unit. Check your IODEVICE statements in the IOCP input data set and make sure that the device types are valid and reflect the true physical device they are defining.

For MVSCP input data sets: HCD supports device types that previously had to be defined as "look-alike" devices for MVSCP. For example, for an IBM 3251 graphic device (previously defined as 2250-3), it is mandatory that the correct device type is defined in the UNIT parameter. HCD validates the PCU parameter, which is different for the IBM 3251 and the IBM 2250-3 graphic devices.

A device whose device type supports the dynamic capability may be defined as dynamic or not dynamic by means of the DYNAMIC parameter. However, there are programs, including customer programs, supplier programs and IBM products, that depend on device related data structures such as UCB and EDT, or use existing operating system services which access these data structures, and are unprepared to handle dynamic changes to these structures.

Therefore, HCD considers devices that are not specified with the DYNAMIC parameter in the IODEVICE statement as "installation-static", as it does if DYNAMIC=NO. That means, that the device might support the dynamic capability, but the installation requests that the device is not treated as dynamic.

Installation-static devices can be dynamically added to the software I/O configuration, but can not be deleted or modified while z/OS is running.

When migrating the MVSCP input data sets, HCD shows **no** value as default with the DYNAMIC and the LOCANY parameter, instead of specifying NO.

For HCPRIO input data sets: HCD supports device types that previously had to be defined as "look-alike" devices for HCPRIO. For example, you can define an IBM 6262 printer device (previously defined as an IBM 4248 printer device in the HCPRIO input data set) with a device type of 6262.

The support for VM type devices has been brought into line with the support for MVS type devices. However, there might be some differences to HCPRIO device type support (for example, concerning the MODEL parameter).

Esoteric token

HCD introduces an esoteric token used during allocation to find the appropriate esoteric for a data set that has been cataloged using the esoteric. You no longer have to maintain a chronological order and may delete and add esoterics without getting access problems for data sets that are cataloged using esoterics.

You may use the HCD profile (see “Defining an HCD profile” on page 19) to tell HCD to assign a token in ascending order to each esoteric when migrating an MVSCP input data set.

If you do not want to assign tokens in ascending order or when you migrate only parts of a configuration using the incremental update function (see “Updating parts of a configuration by migrating input data sets” on page 313), you can use a parameter on the UNITNAME statement. This parameter lets you specify a token for an esoteric to be migrated to HCD, as follows:

```
UNITNAME=...,TOKEN=nnnn
```

nnnn

is a number from 1 to 8999.

You have to specify a token for all esoterics or for none at all. For more information on catalog considerations, refer to “Data sets cataloged with an esoteric device group name” on page 365.

Assembler statements

HCD generates own macro instructions into the logical input data set before processing. This may cause problems with assembler statements you inserted into your input data sets. For example, coding the ISEQ, CSECT, or RMODE instructions may cause a warning message issued by the assembler. As a consequence the migration will be terminated. To avoid this, remove your assembler statements. The generated HCD instructions look as follows:

```
        PRINT OFF
        COPY CBDZPARS
        TITLE 'xxx LISTING'
CBDTXT CSECT
CBDTXT RMODE ANY
        PRINT ON NOGEN
        .
        .
input data set statements
        .
        .
        PRINT OFF
        HCDEND
        END CBDTXT
```

Preparing additional input data sets for migration

This section describes what to consider when migrating more than one MVSCP, IOCP, or HCPRIO input data set into one IODF.

When you migrate additional input data sets into an IODF that already contains definitions, these input data sets may contain control units and devices that are already defined in the existing IODF. HCD assumes that added control units that already exist in the IODF, refer to the same physical control unit, and that the control unit is shared between processors. The following sections describe the rules when a control unit or device is mapped to an existing one and when it is newly defined.

Migrating additional IOCP input data sets

When migrating additional IOCP input data sets, the mapping of control units and devices depends on whether the attached control units are already defined in the IODF or not.

The same control unit number is already defined in the IODF: The control unit is mapped, if both:

- the control unit type is the same, and
- the number and type of attached devices are the same.

If one of these conditions is not fulfilled, the control unit definition is rejected. Figure 126 to Figure 128 on page 284 show examples of how control units are mapped. These examples do not show more than two control units attached to devices, but the same rules also apply if more control units are attached.

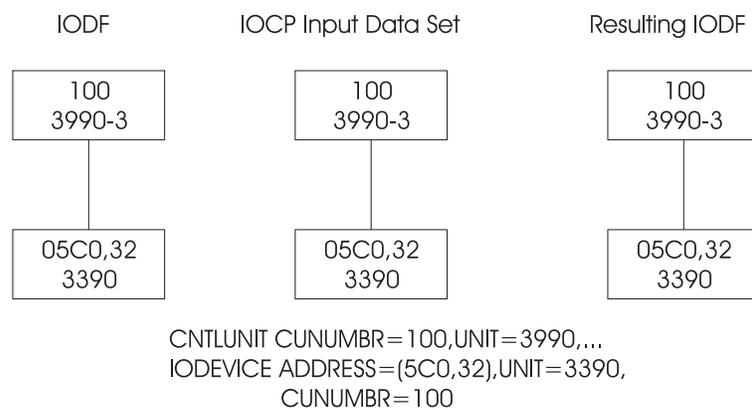


Figure 126. IOCP Migration. Control units and devices are mapped, because they already exist in the IODF.

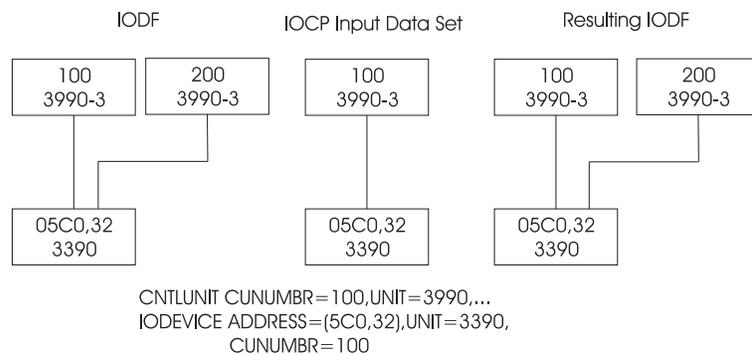


Figure 127. IOCP Migration. Control units and devices are mapped, although the devices also attached to another control unit in the IODF.

The following example shows the same control units in the IODF and IOCP input data set, but the attached devices are shared in the IOCP input data set. In this case, the devices are merged and will be shared after the migration (only if the control units do not connect to the same processor configuration).

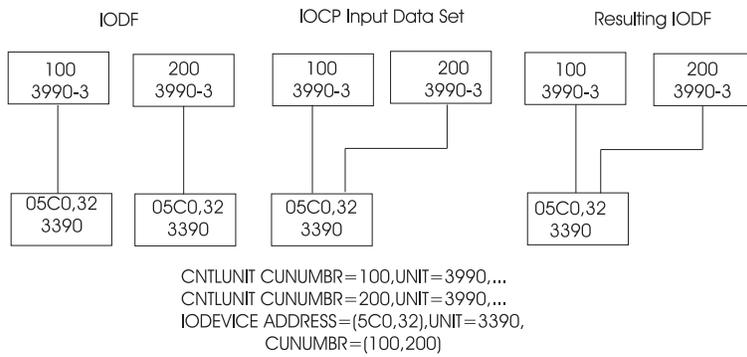


Figure 128. IOCP Migration. Control units and devices are mapped (only if the control units do not connect to the same processor configuration).

The same control unit number is not yet defined in the IODF: In this case:

- a new control unit is defined.
- a new device is defined, unless
 - the device in the input data set is attached to a control unit, to which it is already attached in the IODF. In this case, the new device is mapped to the existing one and attached to both control units (see Figure 129.)
 - or -
 - a device with same device number and type already exists in the IODF and is not attached to any control unit. In this case, the new device is mapped to the existing one and attached to the new control unit (see Figure 130 on page 285).

If none of these conditions is fulfilled, a new device is defined (see Figure 131 on page 285).

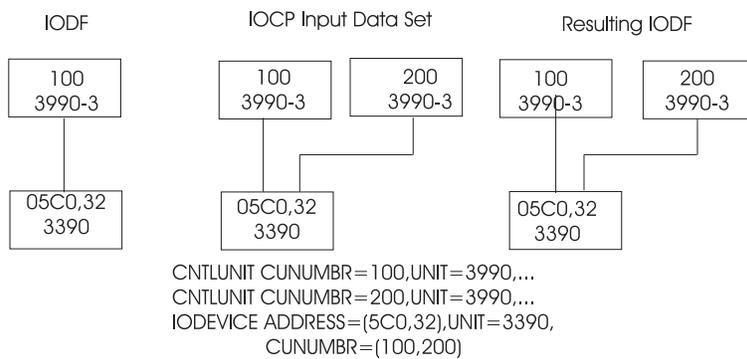


Figure 129. IOCP Migration. Devices are mapped, because one attached control unit is already defined in the IODF.

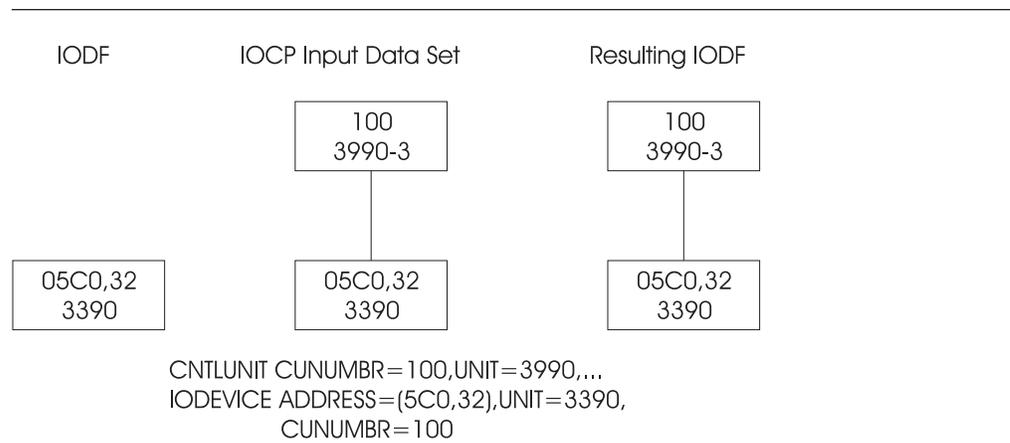


Figure 130. IOCP Migration. Devices are mapped, because the attached control unit is not yet defined in the IODF.

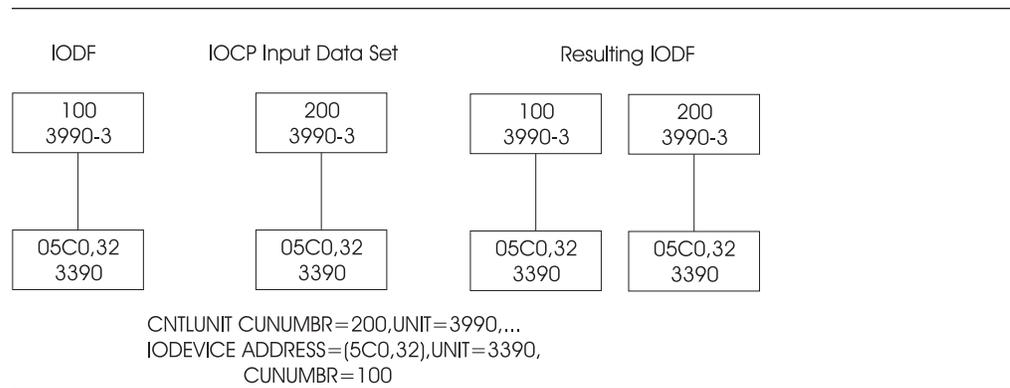


Figure 131. IOCP Migration. Devices are not mapped, because the control unit number does not exist in the IODF.

Migrating additional MVSCP or HCPRIO input data sets

A device can only be mapped if the device number and device type are the same. If the device number or type is not the same, a new device is defined.

If the device number and type are the same, HCD maps the device according to the following rules:

1. If you specify an associated processor and partition on the Migrate IOCP / MVSCP / HCPRIO Data dialog or with the batch migration utility, HCD maps the new device to a device with the same device number and type connected to this processor and partition (provided that such a device exists).
2. If you do not specify an associated processor and partition or the new device does not attach to the specified processor and partition, HCD checks if the device is attached to a control unit. The device in the input data set is mapped, if:
 - the device in the IODF is attached to the same control unit
 - or -
 - the device in the IODF is not attached to any control unit.

This is illustrated in Figure 132 on page 286.

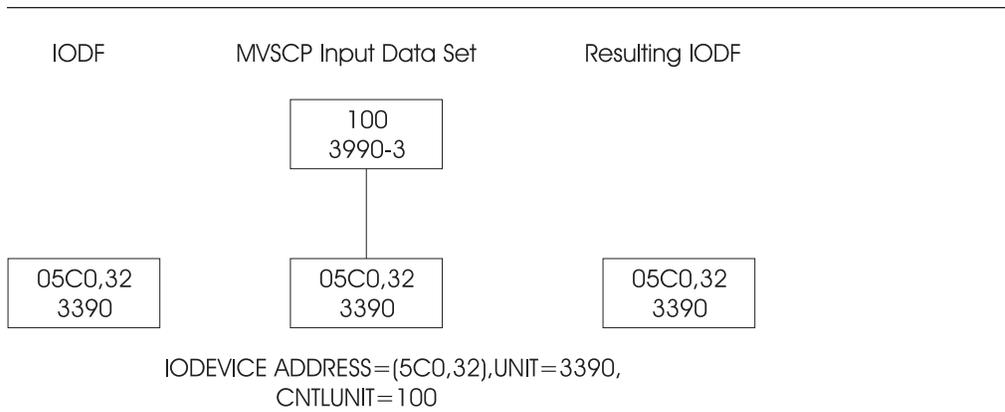


Figure 132. MVSCP Migration. Devices are mapped, because the devices are not attached to a control unit in the IODF.

3. If the new device is not attached to a control unit, the device is mapped to the first device found with the same device number and type.
If the IODF contains several devices with the same device number and type, the device of the MVSCP input data set can be erroneously mapped to a wrong device. To avoid this, you can specify the associated processor and partition on the Migrate IOCP / MVSCP / HCPRIO Data dialog.

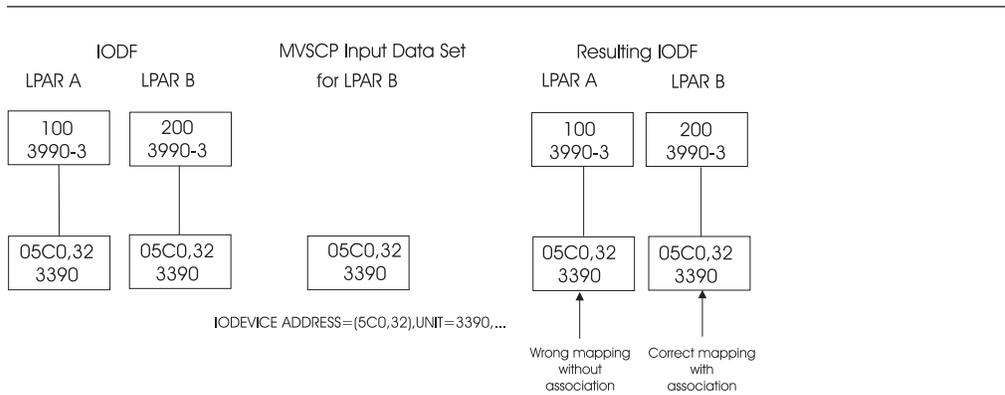


Figure 133. MVSCP Migration. Devices are mapped to the first devices found with the same type and number.

If only a subset of devices in the input data set is already defined in the IODF, this subset is mapped to the existing devices and the remaining new definitions are added.

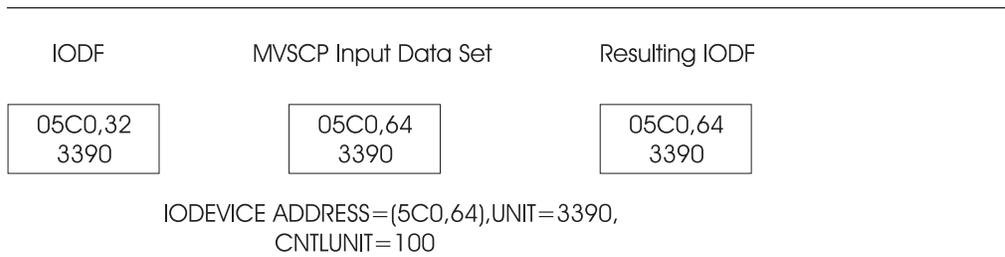


Figure 134. MVSCP Migration. 32 devices are mapped, the other 32 devices are newly defined.

Migrating input data sets using the HCD dialog

The following steps describe how to migrate IOCP, MVSCP, or HCPRIO input data sets to an IODF using the HCD dialog.

Step 1: Specify the work IODF

Before starting the migration, you require a work IODF. You can create a new work IODF or use an existing one.

1. On the *Primary Task Selection* panel enter the name of the IODF to which you want to migrate your input data sets.
2. Select *Migrate configuration data*.
If you create a new work IODF, a dialog appears on which you have to enter IODF specifications. (Refer to Figure 11 on page 33.)
3. From the following *Migrate Configuration Data* menu, select *Migrate IOCP/OS data*. The *Migrate IOCP / MVSCP / HCPRIO Data* dialog shown in Figure 135 appears.

Step 2: Migrate the input data sets

```

Migrate IOCP / MVSCP / HCPRIO Data

Specify or revise the following values.

Processor ID . . . . . RAP2____ + CSS ID . . . . . _ +
OS configuration ID . . . . . _____ +

Combined IOCP/MVSCP input data set . 'HCI.FP.DECKS(FPTST)'  

IOCP only input data set . . . . . _____  

MVSCP only or HCPRIO input data set _____  

Associated with processor _____ +  

partition _____ +

Processing mode . . . . . 2 1. Validate  

2. Save

Migrate options . . . . . 1 1. Complete  

2. Incremental  

3. PCHIDs

MACLIB used . . . . . 'SYS1.MACLIB'  

Volume serial number . . . _____ + (if not cataloged)

```

Figure 135. *Migrate IOCP / MVSCP / HCPRIO Data*

1. Specify the identifier of the processor or operating system with which the input data sets will be associated.
 - For an IOCP input data set migration, specify a processor ID.
 - For an MVSCP or HCPRIO input data set migration, specify an OS configuration ID.

If the specified processor or operating system does not exist in the IODF, a dialog appears that allows you to define a new processor or operating system in the IODF.

Migrating of a single channel subsystem (CSS) to an XMP processor is supported via the incremental migrate option (see Figure 135). Thus, you can consolidate multiple SMP processors on a single XMP processor using the migrate function. When migrating an SMP processor to an XMP processor, you must specify the target CSS. As default, CSS 0 is used.

2. Specify the input data set:

- If you are migrating a combined MVSCP/IOCP input data set, specify the Combined IOCP/MVSCP input data set field.
 - If you are migrating an IOCP input data set only, specify the IOCP only input data set field.
 - If you are migrating an MVSCP or HCPRIO input data set only, specify MVSCP only or HCPRIO input data set field.
 - If you have separate IOCP and MVSCP (or IOCP and HCPRIO) input data sets, but want to migrate both into one IODF, specify both the IOCP only input data set and the MVSCP only or HCPRIO input data set field.
3. The specification of the Associated with processor and partition fields is only applicable if you migrate MVSCP or HCPRIO input data sets to an IODF. Specify a processor and partition with which you want to associate definitions in the input data sets. HCD uses this information to map devices correctly if the IODF contains duplicate device numbers. For more information about this mapping, refer to “Migrating additional MVSCP or HCPRIO input data sets” on page 285.
4. Specify the processing mode:
- **Validate** causes HCD to check the input and to inform you if errors are discovered. HCD *does not store the input* in the IODF, even if the input data set is free of errors. Only the new processor and/or operating system definitions that you defined are stored.
The validate mode provides detailed messages how HCD treats control units and devices that already exist in the IODF.
 - **Save** causes HCD to check the input, and if free of errors, to store the data in the IODF. If there are errors in the input data set(s), HCD informs you by a message, and depending on the severity of the error, does not write the input to the IODF.
5. Specify the migrate option:
- Complete**
Select this option if you want to add a complete processor and/or OS configuration.
- Incremental**
If the specified processor or OS configuration already contains definitions, you can add and replace existing objects with the new information defined in the input data sets. Select this option for this partial migration on the dialog from Figure 135 on page 287. Refer to “Updating parts of a configuration by migrating input data sets” on page 313 for more information about the partial migration.
- PCHIDs**
The migration task also allows updating the PCHIDs of a processor configuration in a validated work IODF with an IOCP input data set that has been generated by the CHPID Mapping Tool. HCD checks that the tokens in the IODF and in the IOCP input data set are matching. For more information on this process, see “How to interact with the CHPID Mapping Tool” on page 224.
6. If the CBDZPARS macro, which contains the migration parsing macros, is not in SYS1.MACLIB, specify the name of the library that contains it. If the library is not cataloged, specify the volume serial number.

After the input has been accepted, HCD issues a message informing you that the migration of input data sets is in process.

Step 3: Analyze errors and correct the input data

During the migration process, HCD first invokes the assembler that parses the input statements. If it detects an error, migration is terminated.

- HCD writes a message to the terminal indicating that migration completed with return code RC=12.
- HCD writes a message to the HCD message list indicating that the assembler completed with a return code other than zero.
- The assembler writes information to the assembly listing that describes the problem in more detail.

If the assembler does not detect any errors, HCD runs a validation check. If it detects an error, HCD writes:

- a message to the terminal indicating that migration completed with a return code higher than 4
- error messages to the HCD message list describing the validation problem.

If the return code is 0 or 4, the IODF is updated and saved (if you specified to save the data). It is, however, recommended that you review the message log. HCD may have made assumptions that are contrary to your configuration requirements. Your actions:

1. Review the message list. For explanations and examples refer to “Resolving migration errors” on page 322.
2. Edit and correct the IOCP, MVSCP, and HCPRIO input data sets.
3. Migrate your input data sets again.

Step 4: Update configuration data

If the protocol or the attached devices specified in the IOCP input data set do not match the supported control unit model, HCD may change the model definitions.

If HCD changes definitions, you are informed by messages. Review the messages, and follow the recommendation provided in the individual message.

If the type/model designated by HCD does not match the real type/model of the control unit, use the HCD dialog to specify the correct type/model.

Also, if this control unit is to be shared with another processor, update the IOCP input data set that is to be migrated accordingly.

Note: The configuration stored in an IODF may not match the IOCP/MVSCP or HCPRIO input. If discrepancies occur, you can make corrections by using the dialog.

The following note only applies if you do not use the extended migration function as described in “Changing I/O configurations by editing data sets” on page 291.

For coupling facility migration: After the IOCP input has been accepted, HCD issues messages informing you that the CF control unit and CF device definitions of the IOCP input data set were ignored.

Migrating input data sets using the batch utility

The following steps explain how to migrate your input data sets using the HCD batch utility instead of the HCD dialog. The batch utility is an easy way to migrate your input data sets if you are not familiar with the dialog yet. You have to use it for migrating your input data sets from MVS/XA SP 2.n or MVS/ESA SP 3.n.

Step 1: Create the work IODF

If an IODF does not yet exist, you first have to create a work IODF into which you want to migrate your MVSCP, IOCP, or HCPRIO data sets. Refer to “Initialize IODF” on page 329 on how to create and initialize an IODF.

Step 2: Migrate input data set

The HCD utility function for migration allows you to migrate the content of MVSCP, IOCP, and HCPRIO input data sets and to store the definitions into an IODF. For a detailed description of the utility function for migration, refer to “Migrate I/O configuration statements” on page 331.

Step 3: Analyze errors and correct the input data

During the migration process, HCD first invokes the assembler that parses the input statements. If it detects an error, the migration process is terminated.

- HCD writes a message to the data set allocated by HCDMLOG that the migration completed with return code RC=12.
- HCD writes a message to the HCD migration log (HCDPRINT) indicating that the assembler completed with a return code other than zero.
- The assembler writes information to the assembly listing (HCDASMP) that describes the problem in more detail.

If the assembler does not detect any errors, HCD runs a validation check. If it detects an error:

- HCD writes a message to the data set allocated by HCDMLOG that the migration completed with return code higher than 4.
- HCD writes error messages to the HCD migration log (HCDPRINT) describing the validation problem.

If the return code is 0 or 4, the IODF is updated. It is, however, recommended that you review the migration log. HCD may have made assumptions that are contrary to your configuration requirements. Your actions:

1. Review the migration log. For explanations and examples refer to “Errors detected during validation process” on page 324.
2. Edit and correct the IOCP, MVSCP, and HCPRIO input data sets.
3. Migrate your input data sets again.

Step 4: Build production IODF

Before you can use the IODF to IPL your operating system you have to convert the work IODF into a production IODF. Refer to “Build a Production IODF” on page 334 for an example on how to build a production IODF using the work IODF.

Changing I/O configurations by editing data sets

With HCD, the extended migration function and the possibilities for writing and migrating configurations allow users to define or change configuration definitions without using the HCD dialog.

Using I/O configuration statements with IOCP/MVSCP syntax, you can also define all configuration objects with their attributes and their connections. Type these statements into a data set as input to the migrate function. For example, a switch together with its ports can be defined via this method.

It is also possible to recreate data sets containing I/O configuration statements for the processor, operating system and switch configurations from an IODF.

However, for some tasks, such as deleting certain configuration objects, it is necessary or easier to make the configuration changes directly in the HCD dialog. For more details on the capabilities of the migration function refer to Table 23 on page 314.

Processor configurations

The migrate IOCP function allows you to specify parameters, additional to IOCP, in the input data set. Analogously, when building an IOCP input data set from the IODF, information is generated which describes the additional parameters of the configuration objects. Processor configuration data sets can be built using the Build IOCP Input Data Set function or the CONFIG PR batch utility. For details, refer to “Build an IOCP input data set” on page 219 and to “Build I/O configuration data” on page 339.

For a description of the complete and valid syntax rules for all IOCP keywords and parameters, refer to the *IOCP User's Guide* for your processor.

Operating system configurations

The migrate MVSCP function allows you to specify all configuration data of an OS configuration, for example the device preference values for esoterics or the user parameters for devices. Also, it is possible to generate an OS configuration data set from the IODF using the Build OS Configuration Data Set dialog or the CONFIG OS batch utility. For details, refer to “Build I/O configuration data” on page 226 and to “Build I/O configuration data” on page 339.

Switch configurations and switch-to-switch connections

It is possible to migrate switch definitions with all ports, switch-to-switch connections and all switch configurations from a data set. Analogously, it is possible to build such a data set from the IODF using the CONFIG SW batch utility. For details, refer to “Build I/O configuration data” on page 339.

Important Note:

It may not be possible to remigrate an IOCP input data set generated by HCD back into the IODF. The reasons are:

- HCD uses the High Level Assembler program for parsing the IOCP statements. The High Level Assembler earlier than V1.5 is restricted to 255 characters for any keyword value. IOCP statements, however, may contain keywords with a value greater than 255 characters. High Level Assembler V1.5 removes this restriction.

- HCD keeps additional data for a processor configuration that is not contained in an IOCP input data set. This data may be used for validation and, therefore, missing at the migrate step leading to validation errors. For example, the partition usage is defaulted to CF/OS. For a shared CF peer channel, this may lead to a validation error, because only a CF partition may be specified in the access or candidate list.
- Since the IOCP data are only a subset of the processor configuration data, you may lose this additional configuration data if you update a processor configuration from an IOCP input data set.
- IOCP data sets do not contain devices connected to a processor with a null device candidate list because of IOCP rules. If the device is connected to another processor with the same control unit, this is an ambiguous configuration and is not migrated.

For updating the IODF via I/O control statements, it is recommended to use the extended I/O configuration statements of HCD instead of an IOCP input data set (see “IOCP input data sets using extended migration” on page 221).

Additional parameters and statements

In the following, the configuration objects and their attributes you can define via I/O control statements and migrate into HCD are described. For a detailed description of the IOCP keywords and parameters, refer to the *IOCP User's Guide* for your processor.

Configuration object	Attributes
Operating system	ID Name Type Description D/R site OS configuration
NIP console	Device number
EDT	ID Preference value Description
Esoteric	Name Device numbers Token VIO indication
Generic	Name Preference value VIO indication
Switch	ID Unit Model Description Serial number Installed ports Chained switch connection

Configuration object	Attributes
Port	Address Name Occupied indication
Switch configuration	ID Switch ID Description Default dynamic connection
Port configuration	ID Allowed dynamic connections Prohibited dynamic connections Dedicated connection Blocked indication
Processor	ID Unit Model Support level Configuration mode SNA address Serial number Description
Channel Subsystem	ID Description Maximum number of devices
Partition	Name Number Usage Description Uniqueness of PCIe Function UIDs
PCIe function	ID Unit Type PCHID CHID (PCHID,VCHID) Description PNET IDs Partition access list Partition candidate list User Defined Identifier (UID)

Configuration object	Attributes
Channel path	ID Type Operation mode Adapter details (PCHID, VCHID, HCA) PNET IDs Partition access list Partition candidate list Channel parameter I/O cluster Dynamic switch Description Coupling facility (CF) connection information Entry switch/port connected to a channel path User Defined Identifier (UID)
Control unit	Number Unit Model Serial number Switch/port connected to a control unit Description Connected channel paths/link addresses Unit address ranges I/O concurrency level Protocol Logical address
Device	Number Unit Model Serial number Volume serial number Connected control units Description Unit address Preferred channel path TIMEOUT indication Status detection indication OS parameters/features Subchannel set ID

Operating system

An operating system is specified with the IOCONFIG statement. It contains the following parameters:

- ID** Specifies the numerical identifier of the OS (mandatory).
 This keyword is maintained for compatibility with the MVSCP syntax.
- NAME** Specifies the OS configuration ID. This is required only, if the migration is performed using the wildcard * in the parameter string of the batch utility. Otherwise, it is ignored and the OS configuration ID is taken from the invocation parameters.
- TYPE** Specifies the OS configuration type. This is required only, if the migration

is performed using the wildcard * in the parameter string of the batch utility. Otherwise, it is ignored and the OS configuration type is taken from the invocation parameters.

DESC Specifies a description of the operating system (optional). The description of the OS configuration is added or updated.

DROSID

Specifies the alphanumeric identifier of the D/R site OS configuration (optional).

Table 7. Operating System Syntax

Syntax:

ID=id	2 alphanumeric characters
NAME=os_name	up to 8 alphanumeric characters
TYPE=type	MVS or VM
DESC='description'	up to 32 characters
DROSID=os_name	up to 8 alphanumeric characters

Example: The following example defines an OS configuration named NEWOS01B of type MVS together with the given description.

```
IOCONFIG ID=01,NAME=NEWOS01B,DESC='LPAR system',TYPE=MVS,DROSID=NEWOSDR1
```

NIP consoles

A NIP console is specified with the NIPCON statement. It contains the DEVNUM keyword.

DEVNUM

Specifies a list of device numbers to be used as NIP consoles (mandatory). All devices specified must be defined in your configuration.

NIP tries to use the devices in the order they are listed (left-to-right).

Table 8. NIP consoles Syntax

Syntax:

DEVNUM=(device_number[,device_number]...)	list of device numbers
---	------------------------

Example: In the example the devices with numbers 102E and 102F are assigned to be used as consoles. NIP will try device 102E first.

```
NIPCON DEVNUM=(102E,102F)
```

EDT

An Eligible Device Table is specified with the EDT statement.

Note: If you want to define only one EDT, place the EDT statement ahead of all UNITNAME statements (see “Esoteric” on page 296 and “Generic” on page 297). If multiple EDT statements are written, each statement must precede the UNITNAME statements defining the corresponding EDT.

The EDT statement contains the following parameters:

ID Specifies the identifier of the EDT. The default is the ID specified on the IOCONFIG statement. If multiple EDT statements are written, the ID parameter is mandatory for all but one statement.

DEVPREF

Specifies a list of devices in the order of preference (optional).

This parameter conforms to legacy syntax but is still supported. However, it is recommended to use the DEVPREF parameter with the UNITNAME statement, see “Generic” on page 297, instead.

DESC Specifies a description of the EDT (optional).

Table 9. EDT Syntax

Syntax:

ID=id	2 hexadecimal characters
DEVPREF=(generic_name[,generic_name]...)	list of generic device types, generates the preference value of generics according to the list position
DESC='description'	up to 32 characters

Example: The following example defines EDT 01 with the given description.

```
EDT ID=01,DESC='Eligible Device Table 1'
```

Esoteric

An esoteric is specified with the UNITNAME statement. For use with esoterics it contains the following parameters:

NAME

Specifies the name of the esoteric (mandatory).

Note: Do not use esoteric names SYSALLDA, SYS3480R, or SYS348XR.

UNIT Specifies a sequence of consecutive device numbers (optional). The numbers specified must be defined in the configuration.

TOKEN

Allows controlling the order of esoterics in the EDT (optional). Only relevant if you have data sets that are cataloged using esoterics.

Tokens prevent the order of esoterics from becoming alphabetical after IPL, thus avoiding access problems for data sets that are cataloged using esoterics.

If a token is specified for one esoteric you must also specify tokens for all other esoterics.

VIO States whether or not the devices are eligible for VIO. May only be set to YES if the esoteric contains at least one DASD device type (optional).

Table 10. Esoteric Syntax

Syntax:

NAME=esoteric_name	8 alphanumeric characters
UNIT=((devnum,n)[,(devnum,n)]...)	four hexadecimal characters followed by a decimal number
TOKEN=token	number in the range 1 to 8999
VIO=value	YES or NO

Example: The following example assigns eight device numbers 01D1 through 01D8 and the token value to esoteric device group ES002. The group is not eligible for VIO.

```

UNITNAME NAME=ES002,          *
          UNIT=((01D1,8)),     *
          TOKEN=2015,         *
          VIO=NO

```

Generic

A generic is specified with the UNITNAME statement. For use with generics it contains the following parameters:

NAME

Specifies the name of the generic (mandatory; must be a valid generic name).

VIO Specifies whether or not the devices are eligible for VIO (optional). VIO=YES may only be specified if the generic name specifies a DASD device type.

DEVPREF

Allocates a position in a preference order of generics (optional).

For default values see the information on MVS devices within the “Supported hardware report” on page 425. This parameter value must be unique for the OS configuration.

Table 11. Generic Syntax

Syntax:

NAME=generic_name	eight alphanumeric characters
VIO=value	YES or NO
DEVPREF=pref_value	decimal number between 1 and 99999

Example: The following example defines the generic device group 3390 with the preference value 150. The devices are eligible for VIO.

```

UNITNAME NAME=3390,          *
          VIO=YES,           *
          DEVPREF=150

```

Switch

A switch is specified with the SWITCH statement. It contains the following parameters:

SWID Specifies an identifier for the switch (mandatory).

UNIT Specifies the switch unit (mandatory).

MODEL

Specifies the switch model (optional).

DESC Specifies a description of the switch (optional).

SERIAL

Specifies a serial number (optional).

PORT Specifies the installed ports (default from UIM, optional).

Specify only ports that are supported by the switch type. At least the minimum installed port range is set to installed. If the switch already exists with an installed port range that differs from the specified installed port range(s), only the new specified ports are set to installed if possible. That

means, any existing installed port that does not hold a connection to a channel path or control unit is set to not installed if not specified with the PORT parameter.

SWPORT

Specifies the chained switch connections (optional). If operand exceeds 255 characters, repeat the SWITCH statement with the remaining values.

If the switch already exists, all existing connections to other switches are broken. The connections to other switches are established as specified by the SWPORT parameter.

ADDRESS

Specifies the switch address for a FICON switch (optional).

The corresponding switch control unit and device are specified by corresponding CNTLUNIT and IODEVICE statements.

Table 12. Switch Syntax

Syntax:

SWID=id	2 hexadecimal characters
UNIT=switch_unit	like CU type
MODEL=switch_model	like CU model
DESC='description'	up to 32 characters
SERIAL=serial_no	up to 10 characters
PORT=((low_port_id,high_port_id),...)	up to 32 port ranges
SWPORT=((from_port,to_switch,to_port),...)	up to 32 switch-to-switch connections
ADDRESS=switch_address	2 hexadecimal characters (domain ID); for FICON switches only

Example: In the following example, switch 20 with an installed port range 00 to FF is connected to port D1 of switch 01 via port C0.

```
SWITCH SWID=20,UNIT=2032, *
      PORT=((00,FF)), *
      DESC='FICON switch, installed 10/09/11', *
      SERIAL=55-8888, *
      SWPORT=(C0,01,D1) *
      ADDRESS=02
```

Note: Put the serial numbers in quotes, if you use characters such as blanks or commas as part of your serial numbers.

Port

A switch port is specified with the PORT statement. It contains the following keywords:

ID Port address identifying the port (mandatory).

NAME

Specifies a port name (optional).

To be accepted within I/O Operations (ESCON Manager) commands the port name must not include commas, asterisks, or blanks. It must not contain X'FF' or any extended binary-coded decimal interchange code

(EBCDIC) character less than X'40'. It must also not begin with a left parenthesis and end with a right parenthesis.

OCC Indicates that the port is connected to a processor, switch, or control unit (optional).

Table 13. Port Syntax

Syntax:

ID=port_id	2 hexadecimal characters
NAME='portname'	up to 24 characters
OCC	no value assigned

Example: In the following example port D5 is named *connected_to_CU_7230* and indicated as occupied.

```
PORT ID=D5, *
      NAME='connected_to_CU_7230', *
      OCC
```

Switch configuration

A switch configuration is specified with the SWCONF statement. It contains the following parameters:

ID Specifies the switch configuration ID (mandatory).

SWID Identifies the switch owning the configuration (mandatory).

DESC Specifies a description of the switch configuration (optional).

DEFCONN

Specifies whether the default port connections are set to allowed or prohibited (mandatory).

Table 14. Switch Configuration Syntax

Syntax:

ID=switch_configuration_id	8 characters
SWID=switch_id	2 hexadecimal characters
DESC='description'	up to 32 characters
DEFCONN=dynamic_default_connection	A (allowed) or P (prohibited)

Example: In the following example the default connection for switch configuration BASECONF of switch 01 is set to allowed.

```
SWCONF ID=BASECONF, *
        SWID=01, *
        DESC='basic configuration', *
        DEFCONN=A
```

Port configuration

A port configuration is specified with the POCONF statement.

The POCONF statement is an optional extension to the switch configuration. With POCONF, port connections are defined explicitly and existing defaults are overridden. The statement may be a maximum of 255 characters long. To express longer statements the POCONF may be repeated. POCONF includes the following parameters:

- ID** Port address identifying the port (mandatory).
- PORTCF**
Specifies the type of connections to target ports (mandatory).
- A** sets the dynamic connection to the succeeding list of target port IDs to allowed
 - P** sets the dynamic connection to the succeeding list of target port IDs to prohibited
 - D** sets a dedicated connection to the succeeding target port ID
- BLOCKED**
blocks the port.

Table 15. Port Configuration Syntax

Syntax:

ID=port_id	2 hexadecimal characters
PORTCF=([A, (id1,...,idn),] [P, (id2,...,idm),] [D, (id),] [BLOCKED])	One or more of the following specifications: list of ports in installed range, dyn. connection allowed list of ports in installed range, dyn. connection prohibited port with dedicated connection BLOCKED attribute

Example: In the following example, port D2 has allowed dynamic connections to ports B1 and B3, and prohibited dynamic connections to B5 and B7. Port D4 has a dedicated connection to port C0.

```
POCONF ID=D2, *
      PORTCF=(A, (B1,B3),P, (B5,B7))
POCONF ID=D4, *
      PORTCF=(D, (C0))
```

Processor

A processor is specified with the ID statement. If specified, it must precede all other statements in the configuration data set. It contains the following keywords:

NAME

Specifies the processor ID. If the ID does not exist, it is created (1).

MSG1 Specifies the identification information that is printed on the ID1 line of the heading in IOCP configuration reports (optional). Only supported for compatibility with IOCP.

MSG2 Specifies the identification information that is printed on the ID2 line of the heading in IOCP configuration reports (optional). Only supported for compatibility with IOCP.

SYSTEM

Specifies the machine limits and rules that IOCP will enforce for a deck verification. The system parameter includes a specification of the processor machine type number and, optionally, a machine limits number (optional). Only supported for compatibility with IOCP.

LSYSTEM

Specifies the system name (CPC designator) of the local system; that is the system which uses this IOCDS.

UNIT Specifies the processor unit (1).

MODEL

Specifies the processor model (1).

LEVEL

Specifies the processor support level (1). For further information on support levels refer to 'support_level_ID'.

SNAADDR

Specifies the SNA address (network name, system name) for a processor in a z Systems cluster (optional).

MODE

Specifies the processor configuration mode as LPAR or BASIC (1).

SERIAL

Specifies the processor serial number (optional).

DESC

Specifies a description for the processor (optional).

Note (1): **UNIT, MODEL and LEVEL** are processed only if the migration is performed using the wildcard * in the parameter string of the batch utility.

Table 16. Processor Syntax

Syntax:

NAME=processor_id	8 characters
MSG1='message'	up to 64 characters; first 8 characters are taken as IOCDS name
MSG2='message'	up to 64 characters
SYSTEM=(processor_unit[,limits_number]	4 characters followed by a decimal number
LSYSTEM=local_cpc_designator	8 characters
UNIT=processor_unit	8 characters
MODEL=processor_model	4 characters
LEVEL=support_level	8 characters
SNAADDR=(network_name,system_name)	list of 2 entries, each up to 8 characters
MODE=processor_mode	BASIC or LPAR
SERIAL=serial_number	up to 10 characters
DESC='description'	up to 32 characters

Example: In the following example processor PROC01 of type 2094, model S28 is defined with the serial number 0518712094, in LPAR mode, support level H050331.

```
ID NAME=PROC01,UNIT=2094,MODEL=S28,          *
   DESC='XMP, Basic 2094 support',SERIAL=0518712094,  *
   MODE=LPAR,LEVEL=H050331
```

Channel subsystem and partition

Depending on whether you use the RESOURCE statement for XMP processors or SMP processors, it specifies the channel subsystems, the logical partitions (names and numbers) and groups the logical partitions to the channel subsystems. It contains the following keywords:

Table 17. Channel subsystem and partition Syntax (continued)

Syntax:

MAXDEV=((CSS(0),maxnum1[,maxnum2, maxnum3]),..., (CSS(n),maxnum1[,maxnum2, maxnum3]))	maximum number of devices for each channel subsystem in each subchannel set
--	--

Example: In the following example, the processor contains three channel subsystems with three partitions in each. Channel subsystem CSS(2) contains two subchannel sets. This is indicated by the MAXDEV statement containing two maximum numbers of devices (35 and 20).

```
RESOURCE PART=((CSS(0),(LP01,1),(LP02,2),(LP03,3)), *
              (CSS(1),(LP11,1),(LP12,2),(LP13,3)), *
              (CSS(2),(LP21,1),(LP22,2),(LP23,3))), *
DESCL=('LPAR1_of_CSS0','LPAR2_of_CSS0','LPAR3_of_CSS0', *
       'LPAR1_of_CSS1','LPAR2_of_CSS1','LPAR3_of_CSS1', *
       'LPAR1_of_CSS2','LPAR2_of_CSS2','LPAR3_of_CSS2'), *
USAGE=(CF/OS,OS,CF,CF/OS,OS,CF,CF/OS,OS,CF), *
MAXDEV=((CSS(0),63),(CSS(1),50),(CSS(2),35,20)), *
CSSDESCL=('first CSS(0)','second CSS(1)','third CSS(2)')
```

For HCD, it is also possible to specify a separate RESOURCE statement for each channel subsystem. This may be required if you do not use a High Level Assembler V1.5 or later. You can split the previous RESOURCE example for an XMP processor into the following parts:

```
RESOURCE PART=((CSS(0),(LP01,1),(LP02,2),(LP03,3))), *
DESCL=('LPAR1_of_CSS0','LPAR2_of_CSS0','LPAR3_of_CSS0'), *
USAGE=(CF/OS,OS,CF), *
MAXDEV=(CSS(0),63), *
CSSDESCL=('first CSS(0)')
RESOURCE PART=((CSS(1),(LP11,1),(LP12,2),(LP13,3))), *
DESCL=('LPAR1_of_CSS1','LPAR2_of_CSS1','LPAR3_of_CSS1'), *
USAGE=(CF/OS,OS,CF), *
MAXDEV=(CSS(1),50), *
CSSDESCL=('second CSS(1)')
RESOURCE PART=((CSS(2),(LP21,1),(LP22,2),(LP23,3))), *
DESCL=('LPAR1_of_CSS2','LPAR2_of_CSS2','LPAR3_of_CSS2'), *
USAGE=(CF/OS,OS,CF), *
MAXDEV=(CSS(2),35,20), *
CSSDESCL=('third CSS(2)')
```

Uniqueness of PCIe User Defined Identifier

A partition may require that all accessing PCIe functions must have distinct User Defined Identifiers (UIDs). This uniqueness-required property of partitions is specified with the UUID statement. The statement is optional and must be given at most once in an input deck.

If given, it specifies a single list of partitions for which uniqueness of UIDs will be checked. The UUID statement contains the following keywords:

PART or PARTITION

Specifies a list of logical partition names that require unique PCIe function UID values that are mandatory. All other (non-listed) partitions of the processor will not have the uniqueness required property. For example, the property is cleared on those other partitions.

| Table 18. PCIe user defined identifier syntax

Syntax:	
UUID PART PARTITION=(* lpname[,lpname2,...])	List of partition names with up to 8 alphanumeric characters that must be specified in the previous RESOURCE statement. A single "*" means that ALL partitions are enabled for UUID checking. A PART or PARTITION keyword is mandatory in a UUID statement. Only one PART or PARTITION keyword is allowed (for example, no multiple lists of partitions).

| **Example:** The following example defines the uniqueness required property for three partitions.

UUID PARTITION=(LP01,LP22,LP24)

PCIe function

A PCIe function is specified with the FUNCTION statement.

The FUNCTION statement contains the following keywords:

- FID** Identifies the PCIe function within the processor configuration (mandatory).
- TYPE** Identifies the PCIe function type (optional - defaults to ROCE).
- UNIT** Identifies the PCIe function type (optional - support for backward compatibility).
- PCHID** Identifies the PCIe adapter card which provides the specified function by specifying the slot of the card in the I/O drawer.
- VCHID** Identifies a virtual PCIe adapter (exclusive to the PCHID keyword)
- VF** Identifies the PCIe virtual function number (optional).
- PNETID** Identifies the physical network IDs.
- PART** Specifies the access and candidate lists of partitions entitled to use the PCIe function.
- DESC** Specifies a description of the PCIe function (optional).
- UID** Specifies a user defined identifier.

| Table 19. PCIe Function Syntax

Syntax:	
FID=xxxx	four hexadecimal characters
TYPE=type	valid PCIe function type
UNIT=type	PCIe function type (use to be backward compatible).
PCHID=xxx	three hexadecimal characters for physical channel id
VCHID=xxx	three hexadecimal characters for virtual channel id
VF=nnn	up to three decimal digits

Table 19. PCIe Function Syntax (continued)

Syntax:

PNETID=(pnetid1,...,pnetid4)	up to four 16-character alphanumeric physical network IDs
PART=(acc_lp1 [, (cand_lp1[, ..., cand_lpn])])	access list (limited to one partition) and optional candidate list of partitions connected to the PCIe function
DESC='PCIe function description'	up to 32 characters
UID=xxxx	up to four hexadecimal digits

Example: The following examples define a PCIe function of type ROCE.

```
FUNCTION FID=020,VF=31,PART=((LP01),(LP21,LP22,LP23,LP24)), *
      TYPE=ROCE,DESC='max VF and 2 pnetid,a=lp01', *
      PNETID=(PNETID1,ID2,,),PCHID=4FE
```

The following example defines a PCIe function of type ISM with UID ABCD.

```
FUNCTION FID=0002,VF=7,PART=((LP01),(=)),TYPE=ISM, *
      PNETID=PNETID2,VCHID=7C0,UID=ABCD
```

Channel path

A channel path is specified with the CHPID statement. For an XMP processor, a spanned CHPID is defined for one or more channel subsystems. Therefore, the CHPID statement must contain the appropriate channel subsystem ID(s). If only one CSS is defined, it is not required to specify its ID on the PATH keyword.

The CHPID statement contains the following keywords:

PATH Specifies the CHPID number and, if required, the CSS IDs of the channel path (mandatory).

PNETID Identifies the physical networks to which the channel path is assigned (optional, only for channel paths of type IQD or OSD).

TYPE Specifies the channel path type of I/O operation for the channel path (mandatory).

SHARED Specifies that the channel path on the CHPID statement is shared (optional).

REC Specifies that the channel path on the CHPID statement is reconfigurable (optional).

PART, PARTITION, or NOTPART

PART and PARTITION specify the access list of logical partitions that will have the channel path configured online after POR, and the candidate list identifying the logical partitions which can access the device.

NOTPART specifies the access list of logical partitions that will not have the channel path configured online after POR, and the list of logical partitions which cannot access the device.

REC REC in the PART or PARTITION keyword allows the channel path to be dynamically moved between partitions after POR.

CHPARAM, OS

Specifies channel path data that is used by the operating system. Examples:

- CHPARAM=00 is the default.
- CHPARAM=01 indicates that the channel path is managed by DCM.

- CHPARM=02 indicates
 - for IQD channels: channel can access the IEDN (IQDX function)
 - for OSD, OSM, and OSX channels: priority queuing is disabled.
 For OSM channel paths, device priority queuing needs to be disabled. Therefore, for these CHPIDs, HCD converts a CHPARM=00 (default with priority queuing enabled) to CHPARM=02.
- CHPARM=40 indicates that the maximum frame size for an IQD channel is 24K.

For more information about the CHPARM parameter, refer to the *IOCP User's Guide* for your processor.

I/O CLUSTER

Specifies an I/O cluster name. An I/O cluster is a sysplex that owns a managed channel path for an LPAR processor configuration.

SWITCH

Specifies a number for a switch which is used as a dynamic switch for all paths from the channel path (CHPID) to the connected control units (required for dynamic connections through a switch).

DESC Specifies a description of the channel path (optional).

TPATH

Specifies a connected CF channel path (optional).

The TPATH parameter can be specified for a CF channel path, either CF receiver, CF sender, or CF peer channel path.

For a CF sender or CF receiver channel path, the TPATH parameter must contain:

- the target processor
- the target CSS ID for XMP processors
- the target channel path ID
- when specified with a CF sender channel path ID, the TPATH parameter must also contain the CF sender control unit and device numbers used for the CF connection (optional for a CF receiver channel paths)

For a CF peer channel path, the TPATH parameter contains the following items for the target and source channel paths of the CF connection:

- the target/source processor
- the target/source CSS ID for XMP processors
- the target/source channel path ID
- control unit number (only if this is a sending CF channel, that is, the channel path connects to a target CF logical partition)
- starting device number (only if this is a sending CF channel)

A CF connection uses two (CF sender channel) or seven (sending CF peer channel) devices. Only the starting one can be specified (with four digits). The remaining devices are automatically assigned to the next consecutive device numbers.

Any CF control units and CF devices specified via CNTLUNIT and IODEVICE statements are ignored.

The connection can only be established if the target channel path exists. If the target channel path is already connected, the existing connection is broken and a new connection is established.

- SWPORT** Specifies an entry switch port (optional).
- PCHID** Specifies a physical channel identifier (optional).
- VCHID** Specifies a virtual channel identifier (optional). Used for internal channel path - exclusiv to the PCHID keyword.
- AID** Specifies the ID of the host communication adapter (HCA) on which the channel is defined.
- PORT** Specifies the port on the HCA on which the channel is defined.

Table 20. Channel Path Syntax

Syntax:

PATH=[(CSS(n,...),]chpid_number[]]	2 hexadecimal characters for the CHPID, and a decimal number for the CSS ID(s)
PNETID=(pnetid1,...,pnetid4)	up to four 16-character alphanumeric physical network ID names
TYPE=type	valid channel path type
SHARED	no value assigned
REC	no value assigned
For XMP processors: PART[ITION]=((CSS(0),(acc_1p1,...,acc_1pn) [, (cand_1p1,...,cand_1p2)] [,REC]) ... [(CSS(n),(acc_1p1,...,acc_1pn) [, (cand_1p1,...,cand_1p2)] [,REC]]))	access list and optional candidate list of partitions; optional addition of REC; for XMP processors: CSS ID is required
For SMP processors: PART[ITION]=((acc_1p1,...,acc_1pn) [, (cand_1p1,...,cand_1p2)] [,REC]))	
For XMP processors: NOTPART=((CSS(0),(acc_1p1,...,acc_1pn) [, (cand_1p1,...,cand_1p2)]) ... [(CSS(n),(acc_1p1,...,acc_1pn) [, (cand_1p1,...,cand_1p2)])]))	up to 2 lists of partitions
For SMP processors: NOTPART=((acc_1p1,...,acc_1pn) [, (cand_1p1,...,cand_1p2)]))	
OS=xx, CHPARM=xx	2 hex character OS parameter
IOCLUSTER=sysplex	8 character sysplex name for managed CHPID
SWITCH=xx	2 hexadecimal characters
DESC='description'	up to 32 characters

Table 20. Channel Path Syntax (continued)

Syntax:

TPATH=(proc,chpid[,CFS CU,CFS device]) (for legacy CF channel path only)	target CHPID for connected CHPID pairs (CF connection): 8 character processor name 2 hex character CHPID 4 hex character CU number 4 hex character device number
TPATH=((proc,chpid[,CFP CU,CFP device]), (proc,chpid[,CFP CU,CFP device])) (for CF peer channel path only)	pair of target and source CHPIDs (CF peer connection): 8 character processor name 2 hex character CHPID 4 hex character CU number 4 hex character device number
TPATH=((CSS(n),proc,chpid[,cu,device]),...)	for XMP processors
SWPORT=(swid,port)	switch and port to which the CHPID connects
PCHID=xxx	three hexadecimal characters for the physical channel ID
VCHID=xxx	three hexadecimal characters for the virtual channel ID
AID=xx	two hexadecimal characters
PORT=n	one numeric character

Examples for a XMP processors: In the following example, the spanned channel path 33 of type IQD is shared by partitions from channel subsystems 0 and 1.

```
CHPID PATH=(CSS(0,1),33),          *
      TYPE=IQD,                    *
      PART=((CSS(0),(LP01,LP02)),   *
            (CSS(1),(LP11,LP12)))
```

Control unit

A control unit is specified with the CNTLUNIT statement. It contains the following keywords:

CUNUMBR

Specifies a number assigned to the control unit (mandatory). The number assigned to each control unit must be unique within an IODE.

UNIT Specifies the type of control unit (mandatory).

SERIAL

Specifies a serial number (optional).

SWPORT

Specifies switch ports to which the control unit is connected (optional). If operand exceeds 255 characters, repeat the CNTLUNIT statement with the remaining values.

DESC Specifies a description of the control unit (optional).

PATH For each channel subsystem, this keyword specifies the channel paths attached to the control unit (mandatory). For control units that are not connected to a processor, specify PATH=** or PATH=(**).

LINK For each channel subsystem, this keyword specifies the link address to which the control unit is attached (optional).

The order in which the link addresses are specified corresponds to the order in which the channel paths are specified in the PATH keyword.

UNITADD

Specifies the unit address ranges that are recognized by the control unit (mandatory).

CUADD

Specifies the logical address for the control unit (optional).

SHARED

Specifies the level of concurrency of I/O requests that the parallel channel path allows for the control unit (optional).

PROTOCOL

Specifies the interface protocol that the parallel control unit uses when operating with the channel paths specified in the PATH keyword (optional).

Table 21. Control Unit Syntax

Syntax:

CUNUMBR=number	4 hexadecimal characters
UNIT=type	valid control unit type
SERIAL=serial_number	up to 10 characters
SWPORT=((swid1,port1),(swid2,port2),...)	list of up to 32 sublists (switch ID, port ID) 2 hex character switch ID 2 hex character port ID
DESC='description'	up to 32 characters max.
For XMP processors: PATH=((CSS(0),chpid[,chpid,...]) [, (CSS(1),chpid[,chpid,...])] ... [, (CSS(n),chpid[,chpid,...])])	0,1,... for the CSS ID; 2 hexadecimal characters for each static CHPID ** for each managed CHPID or for control units not connected to a processor
For SMP processors: PATH=(chpid[,chpid]...)	
For XMP processors: LINK=((CSS(0),link_addr[,link_addr,...]) [, (CSS(1),link_addr[,link_addr,...])] ... [, (CSS(n),link_addr[,link_addr,...])])	0,1,... for the CSS ID; two or four hexadecimal characters for the link address of each CHPID *** if not specified
For SMP processors: LINK=(link_addr[,link_addr]...)	
UNITADD=((address[,number]),...)	2 hexadecimal characters for each unit address followed by a decimal number
CUADD=address	1 or 2 hexadecimal characters
SHARED=value	Y or N
PROTOCOL=value	D, S, or S4

Example for an XMP processor: In the following example, control unit 0780 of type 2105 is connected to channels 11 and 14 in channel subsystem 0 and to channels 21 and 24 in channel subsystem 1. All channels in both channel subsystems use link address E8. Channels from CSS 0 are connected to the control unit via switch 01, channels from CSS are connected via switch 02.

```

CNTLUNIT CUNUMBR=0780,PATH=((CSS(0),11,14),(CSS(1),21,24)), *
UNITADD=((00,128)),LINK=((CSS(0),E8,E8),(CSS(1),E8,E8)),*
CUADD=F,UNIT=2105
DESC='ESS12 780 CU F (3390-mix) VSE128'
SWPORT=((01,E8),(02,E8))

```

Device

A device is specified with the IODEVICE statement. It contains the following keywords:

ADDRESS

Specifies the device number and how many devices are to be defined (mandatory).

UNIT Specifies the device type (mandatory).

MODEL

Specifies the model number of the device, if available (optional).

PART, PARTITION or NOTPART

PART and PARTITION specify the candidate list identifying the logical partitions which can access the device (optional).

NOTPART specifies the logical partitions which cannot access the device (optional).

If for an XMP processors the device has access to more than one CSS, the CSS subkeyword is required to indicate to which channel subsystem the partition belongs.

SERIAL

Specifies the serial number of the device (optional).

VOLSER

Specifies the volume serial number (optional).

CUNUMBR

Specifies the number(s) of the control unit(s) the device is attached to (mandatory).

DESC Specifies a description of the device (optional).

UNITADD

Specifies the unit address that is transmitted on the channel path to select the I/O device (optional). If not specified, the last two digits of the device number are used.

PATH Specifies the preferred channel path (optional).

TIMEOUT

Specifies whether the I/O interface timeout function is to be active (optional).

STADET

Specifies whether the Status Verification Facility is to be enabled or disabled (optional).

SCHSET

Specifies for z9 EC processors or later models the subchannel set ID where the device is located.

PPRCUSE

Specifies the PPRC usage for DASD devices (optional).

Table 22. Device Syntax

Syntax:

ADDRESS=(device_number,number_of_units)	4 hexadecimal characters followed by a decimal number in the range 1 to 4095
UNIT=device_type	up to 8 alphanumeric characters
MODEL=model_number	up to 8 alphanumeric characters
For XMP processors: PART[ITION]=((CSS(0),(lpname1[,lpname2,...])) ... [, (CSS(m),0)] ... [, (CSS(n),(lpname1[,lpname2,...]))])	list of partition names with up to 8 alphanumeric characters; 0,1,... for the CSS ID; For XMP processors, a 0 is accepted as lpname to indicate a null device candidate list; that is, the control unit shared by several CSSs cannot access the device through CSS(m).
For SMP processors: PART[ITION]=(lpname1[,lpname2,...])	
For XMP processors: NOTPART=((CSS(0),(lpname1[,lpname2,...])) ... [, (CSS(m),0)] ... [, (CSS(n),(lpname1[,lpname2,...]))])	list of partition names with up to 8 alphanumeric characters;
For SMP processors: NOTPART=(lpname1[,lpname2,...])	
SERIAL=serial_number	up to 10 numeric characters Note: Put the serial numbers in quotes if you use characters such as blanks or commas.
VOLSER=volume_serial_number	up to 6 characters
CUNUMBR=(number1[,number2]...)	up to 8 hexadecimal numbers of 4 characters (or **** for unconnected devices)
DESC='description'	up to 32 alphanumeric characters
UNITADD=unit_address	2 hexadecimal characters
For XMP processors: PATH=((CSS(0),chpid[,chpid,...]) [, (CSS(1),chpid[,chpid,...])] ... [, (CSS(n),chpid[,chpid,...])])	0,1,... for the CSS ID; 2 hexadecimal characters for each CHPID
For SMP processors: PATH=(chpid[,chpid]...)	
TIMEOUT=value	Y or N
STADET=value	Y or N
SCHSET=n SCHSET=((CSS(0),n),..., (CSS(m),n))	n can be one of 0, 1, or 2. Use the short form SCHSET=n if the placement of the device is the same for all CSSs. SCHSET=0 is the default.
PPRCUSE=value	D (Duplex), F (Flashcopy), S (Simplex), U (Utility), N (Nonsysplex).

Table 22. Device Syntax (continued)

Syntax:

USERPRM=((param1,value1)[,(param2,value2)]...)	list of device specific parameter/value pairs
FEATURES=(feature1[,feature2]...)	list of device specific features
ADAPTER=adapter	up to 5 alphanumeric characters
DYNAMIC=value	Y or N
LOCANY=value	Y or N
NUMSECT=number	decimal number
OFFLINE=value	Y or N
OWNER=value	VTAM or OTHER
PCU=number	decimal value in the range 1 to 4095
SETADDR=value	0, 1, 2, or 3
TCU=value	2701, 2702, or 2703

OS parameters/features: In the following section device specific parameters and features are described. To find out which parameters, private parameters, and features are available to you for a particular device run your Supported HW Report and I/O Definition Reference. (See “Print configuration reports” on page 343 for details on how to run the report function.) Examples of a Supported HW Report and of an I/O Definition Reference are shown in “Supported hardware report” on page 425 and in “I/O Definition Reference” on page 451, respectively.

USERPRM

Allows the specification of OS private parameters.

To locate the private parameters available to you for a particular device refer to your Supported HW Report. The OS private parameters are listed in column SUPPORTED PARAMETERS / FEATURES, following PRIVATE: and extending to the slash (/). For example, device 3590 listed in the sample Supported Hardware Report - MVS Devices (“Supported hardware report” on page 425) supports the private parameters LIBRARY and AUTOSWITCH.

FEATURE

If included in the device specific parameters, FEATURE allows device specific features to be assigned. The features available depend on your UIMs.

To locate the features available to you for a particular device type refer to your Supported HW Report. The features are listed in column SUPPORTED PARAMETERS / FEATURES, following the slash (/). For example, device 3590 listed in the sample Supported Hardware Report - MVS Devices (“Supported hardware report” on page 425) supports the features SHARABLE and COMPACT.

To locate all FEATURE-related information about the device (i.e., possible values, default and description) refer to the 'I/O Definition Reference' report. The report can be written using the HCD batch utility (report type Y).

Note: Device features must be explicitly defined in I/O configuration statements. Otherwise the feature will not be defined during migration.

ADAPTER

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

DYNAMIC

Specifies if the device is eligible for dynamic I/O configuration (optional).

LOCANY

Specifies if UCB can reside in 31 bit storage (optional).

NUMSECT

Specifies the number of guaranteed 256-byte buffer sections in a 2840 display-control buffer allocated to a device 2250-3 (optional).

OFFLINE

Specifies if the device is considered online or offline at IPL (optional).

OWNER

Specifies the subsystem or access method using the device (optional).

PCU Only applicable to a display device 2250-3 attached to a control unit 2840-2 (optional)

Identifies the 2840-2 control unit the 2250-3 display device is attached to. For all 2250-3 devices attached to the same control unit, the same value is specified.

If coded, no separate IODEVICE statement UNIT=2840 must be used.

SETADDR

Specifies which of the 4 set address (SAD) commands is to be issued to the transmission control unit for operations on the line specified by the ADDRESS operand (optional).

TCU Specifies the transmission control unit for a telecommunications line (optional).

Example for an XMP processor: In the following example, the devices numbers 7400 to 741F of type 3390A are defined together with an explicit candidate list: they can be accessed by partition TRX1 from channel subsystem 0 and from partition TRX2 from channel subsystem 1. The Status Verification Facility is enabled. The devices are placed in subchannel set 0 for channel subsystem 0 (this is the default and needs not be specified) and in subchannel set 1 for channel subsystem 1.

```
IODEVICE ADDRESS=(7400,032),UNITADD=50,CUNUMBR=(7300),      *
          STADET=Y,PARTITION=((CSS(0),TRX1),(CSS(1),TRX2)),    *
          SCHSET=((CSS(1),1)),UNIT=3390A
```

Updating parts of a configuration by migrating input data sets

The HCD incremental update function allows you to modify objects in your IODF by specifying the objects with I/O control statements in data sets, e. g. IOCP, MVSCP, or HCPRIO input data sets, and migrating these input data sets into your existing IODF.

Possible actions on objects using the incremental update

Table 23 shows what actions you can perform on objects by using the HCD incremental update function. The meaning of the markup is as follows:

- x Action possible, you can make the change.
- Action not possible, you cannot make the change with the batch migration utility but must use the HCD dialog.
- (x) Action possible. These attributes are deleted when the object itself is deleted.
- (—) Action not possible. These attributes cannot be deleted because the object itself cannot be deleted.

See the notes below the table for further instructions on how to add, delete, or change certain objects, their attributes and connections.

Table 23. Actions on IODF Objects

Object/Attributes	Add	Delete	Change
Processor			
ID	x	—	—
Type/model	x	(—)	—
Support level	x	(—)	—
Configuration mode	x	(—)	—
Serial number	x	—	x
Description	x	—	x
SNA address	x	—	x
Channel Subsystem			
ID	x	—	—
Maximum Number Devices	x	—	x
Description	x	x	x
Partition			
Name	x	x 1)	x 11)
Image number	x	(x)	x11)
Usage type	x 10)	(x)	x
Description	x	(x)	x
UUID			
Partition	x	(x)	x
PCIe function			
FID	x	—	—
VF	x	—	—
UNIT / TYPE	x	(x)	x
CHID (PCHID,VCHID)	x	x	x
PNETID	x	x	x
Partition	x	x	x
UID	x	x	x
Description	x	x	x
Channel path			
CHPID	x	—	—
CHID (PCHID,VCHID)	x	x	x 5)
PNETID	x	x	x 5)
Type	x	(—)	x 5)
Operation mode	x	(—)	x 5)
Access list	x	x	x 5)
Candidate list	x	x	x 5)
Dynamic switch	x	x	x 5)
Switch connection	x	x	x 5; 15)

Table 23. Actions on IODF Objects (continued)

Object/Attributes	Add	Delete	Change
CF connection	x	x	x 5)
Description	x	x	x 5)
OS parameter	x	x	x 5)
I/O cluster	x	x	x 5)
Control unit			
Number	x	x 2)	x 4)
Unit/model	x	(x)	x
Description	x	x	x
Serial number	x	x	x
Switch connection	x	x	x 15)
Channel paths	x	(x)	x
DLA	x	x	x
Logical address (CUADD)	x	x	x
Unit addresses	x	(x)	x
Protocol	x	x	x
IOCL	x	x	x
Device			
Number	x	x 3)	x 4)
Unit/model	x	(x)	x
Description	x	x	x
Serial number	x	x	x
Volume serial number	x	x	x
Control units	x	(x)	x
Processor connect	x	(x)	x
Unit address	x	(x)	x
Preferred CHPID	x	x	x
TIMEOUT	x	x	x
STADET	x	x	x
Candidate list	x	x	x
OS connect	x	—	x
Subchannel set	x	(x)	x
Parameters	x	x	x
Features	x	x	x
User parameters	x	x	x
PPRC usage	x	(x)	x
Operating system			
ID	x	—	—
Type	x	(—)	—
Description	x	—	x
D/R site OS	x	—	x
EDT			
ID	x	—	—
Description	x	—	x
Esoteric			
Name	x	—	—
VIO	x	(—)	x
Device list	x	x 6)	x 7)
Token	x	—	x
Generic			
VIO	x	(—)	x
Preference value	x	(—)	x
Console			
Device list	x	x 9)	x 13)
Order	(x)	(x)	x

Table 23. Actions on IODF Objects (continued)

Object/Attributes	Add	Delete	Change
Switch			
ID	x	—	—
Unit/model	x	(—)	x
Ports (installed range)	x	(—)	x 8)
Serial number	x	—	x
Description	x	—	x
Switch connection	x	x 14)	x 15)
Address	x	—	x
Ports			
ID	x	—	—
Name	x	(—)	x
Occupied indicator	x 15)	x 15)	n/a
Switch configuration			
ID	x	—	—
Description	x	—	x
Default connection	x	—	x
Port configuration			
Allowed connection	x	—	x
Prohibited connection	x	—	x
Dedicated connection	x	—	x
Blocked indicator	x	x	n/a

References to IOCP and MVSCP in the following notes refer to data sets with extended syntax as described in “Changing I/O configurations by editing data sets” on page 291.

Note:

1. To delete a partition, specify all connected channel paths (defined via access or candidate lists) together with their attached I/O units without referring to the partition and without repeating the partition in the RESOURCE statement.

For incremental migration it is recommended to use notpart only in combination with resource statements.

For example: IODF contains 3 LPARS (LpA,LpB and LpC) and CHPID 11 connected to LpA and LpB (nothing else). When migrating a deck with CHPID 11 and a notpart statement to LpC and part statement to LpA, following happens:

- HCD disconnects CHPID 11 from LpA and LpB.
- LpA and LpB will be deleted if they don't have any chpids attached and if no resource statement is given.
- CHPID 11 is deleted and added again.
- HCD now tries to connect to LPARS: In the CHPID statement LpA and LpC are referenced. If they do not exist, they will be created and CHPID 11 will be connected to LpA.

As a result LpB is now deleted, LpC is left untouched.

Note: If only notpart statements would be in the deck, both LpA and LpB would remain deleted, no LPARS could be connected and CHPID 11 could not be created.

2. A control unit is implicitly deleted, if its channel paths are respecified in the IOCP input data set together with their attached I/O units (except the control unit) and it no longer has any connection to a processor.

3. A device is implicitly deleted, if its attaching control units are implicitly deleted using the incremental update specification, and there is no additional connection left to any control unit or operating system.
4. To change control unit and device attributes, specify the entire logical control unit(s) (LCU) the control unit or device is part of. For migration to an OS configuration only (MVSCP data set), respecify the I/O device with the changed parameters. HCD will redefine the device in the corresponding subchannel set.
5. To change channel path attributes, specify all logical control units the channel path is connected to in the IOCP input data set. Otherwise, the channel path is disconnected from the corresponding control units.
6. To delete an esoteric device list, specify all device definitions in the MVSCP input data set, but do not connect them to the esoteric name.
7. To add a device to an esoteric device list, specify the esoteric with the device number.
To delete a device from the esoteric device list, specify the device in an IODEVICE statement) but do not specify the device number for the esoteric device list.
8. An installed port can be set to uninstalled only if it does not belong to the minimum installed port range and does not hold a connection to a channel path or control unit.
9. To delete a console list, specify all contained devices using the IODEVICE statement but do not include the devices in a NIPCON statement.
10. If the usage type is not specified and you add a partition, the usage type is automatically defined: if the IOCP input data set contains a CF receiver channel path with the partition in its access or candidate list, the usage type is set to CF/OS, if not, the usage type is set to OS.
11. To change the partition name or partition number specify the whole partition configuration including all channel paths with attached I/O units which have the partition in their access and candidate lists.
12. (No longer used.)
13. To change a complete console device list, use the NIPCON statement.
To remove a single device from the console list, specify the corresponding IODEVICE statement and omit the NIPCON statement.
14. To delete a switch-to-switch connection, specify a switch via a SWITCH statement and omit the switch-to-switch connection in the SWPORT parameter.
15. When updating switch ports new connections always overwrite a previous connection or status. To be updated with an occupied status the port must currently not be connected.
16. To remove the unique UID requirements for a partition, you need to code a complete UUID statement. The statement needs to reference all partitions that still need to have this capability defined. Removing the unique UID requirement from all partitions of a processor is not possible using this path.

How to invoke the incremental update

1. Specify your objects with IOCP, MVSCP, or HCPRIO control statements. Note that you can add additional parameters and SWITCH statements to exploit the extended migration as described under “Changing I/O configurations by editing data sets” on page 291).

2. Select *Migrate configuration data* on the *Primary Task Selection* panel and on the resulting screen select the *Migrate IOCP/IOS data* option.
3. On the following Migrate IOCP / MVSCP / HCPRIO Data dialog (see Figure 135 on page 287), enter the required data and change the incremental update to Yes.
4. After the input has been accepted, HCD issues a message informing you that the migration of input data sets is in process.

Example 1: Adding a partition

In this example, you specify a partition in your input data set that does not yet exist in the IODF. The attached control units and devices are already defined in the IODF for another processor.

The following figure illustrates the result after the incremental update:

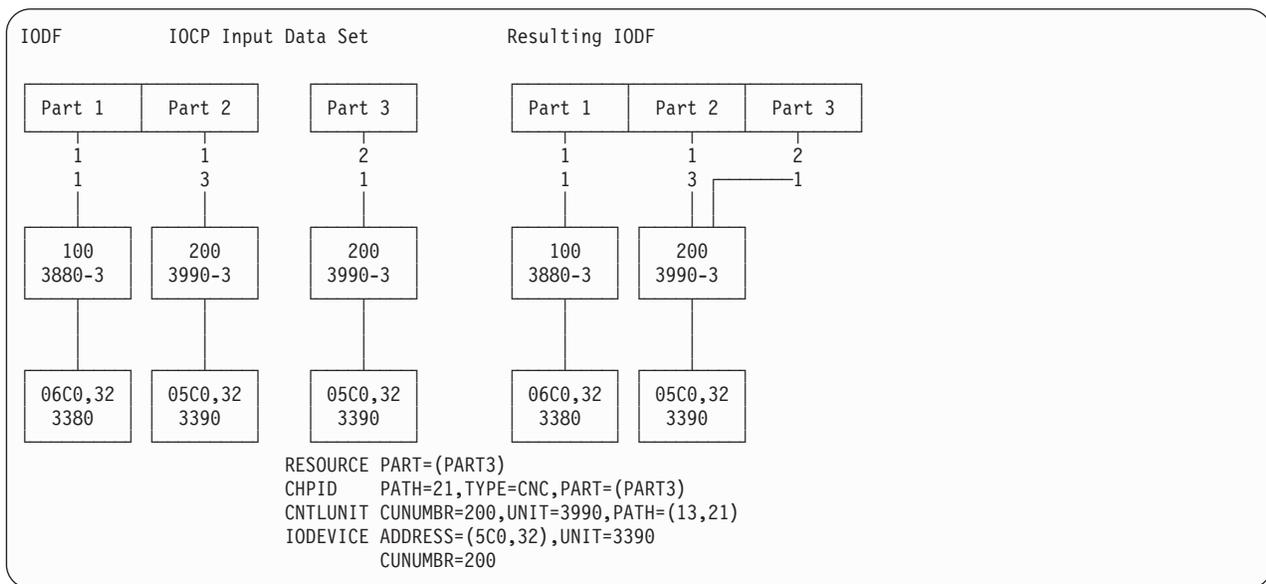


Figure 136. Partial Migration of an IOCP Input Data Set. A new partition is added. Control units and devices are mapped.

The partition is added and the control unit and devices are mapped. For a detailed description when control units and devices are mapped, refer to “Migrating additional IOCP input data sets” on page 283.

Example 2: Replacing a channel path and attached control unit

In this example, you specify a channel path with attached control unit and devices in an IOCP input data set. The channel path, the control unit, and one device already exist in the IODF.

The following figure illustrates the result after the incremental update:

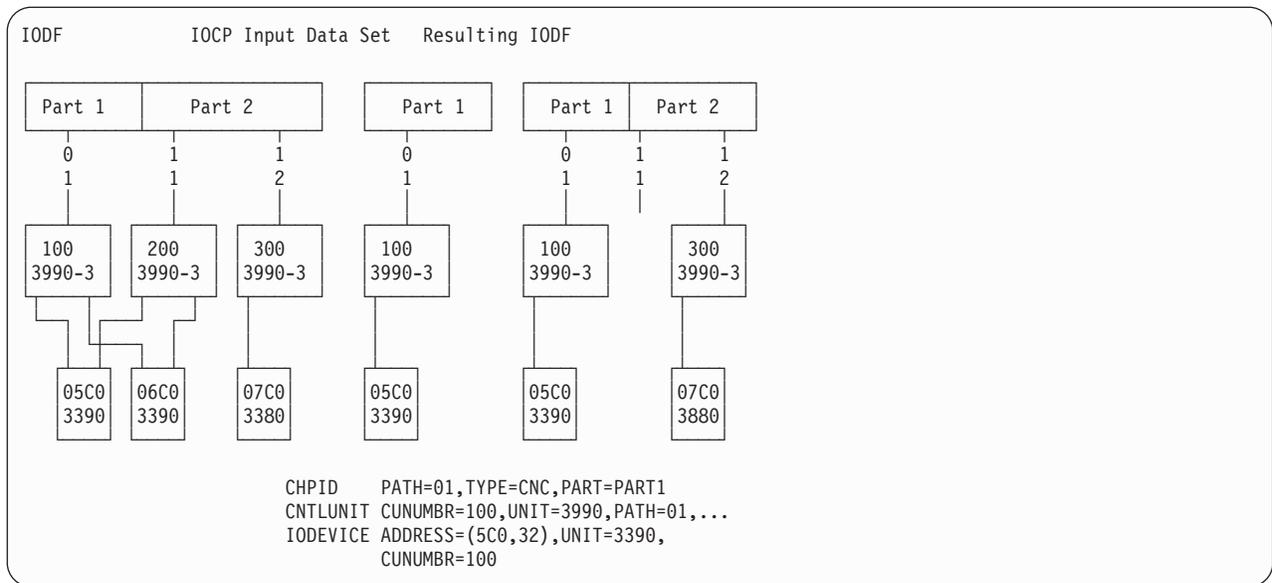


Figure 137. Partial migration of an IOCP input data set. The whole logical control unit (control unit 100 and 200 and connected devices) are replaced by control unit 100 and its connected devices.

The logical control unit in the IOCP input data set replaces the whole logical control unit in the IODF.

Example 3: Replacing a channel path with a new control unit

In this example, you specify channel paths with control unit and device in the input data set. The channel paths are already defined in the IODF, but connect to another control unit.

The following figure illustrates the result after the incremental update:

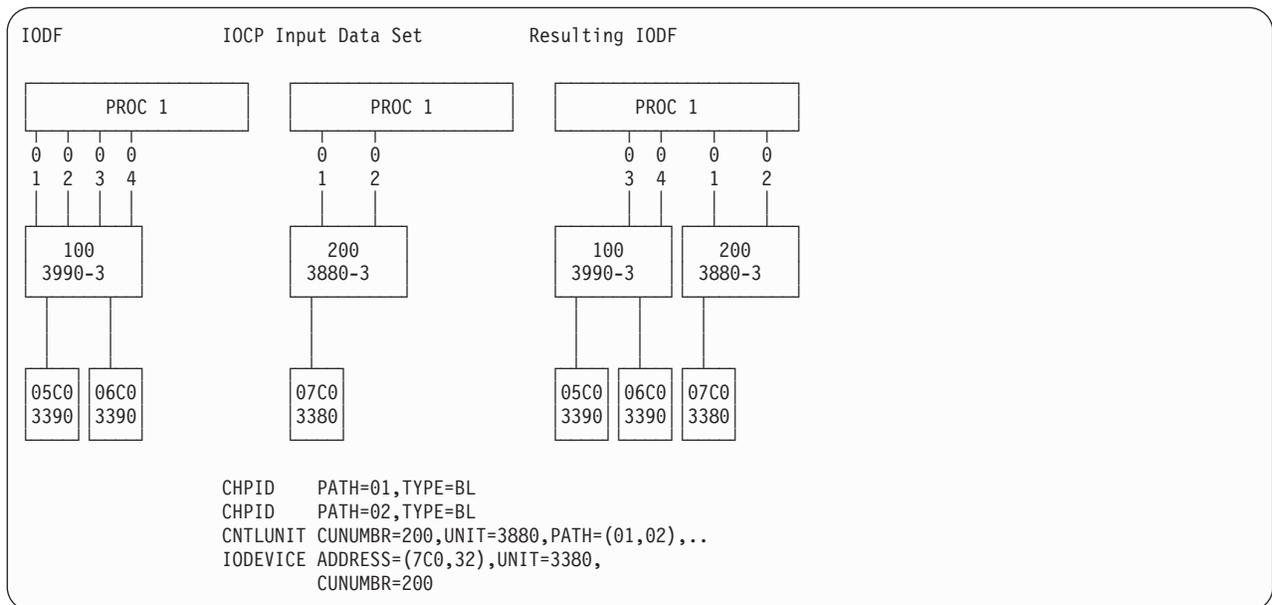


Figure 138. Partial migration of an IOCP input data set. CHPIDs are defined as in the IOCP input data set.

The channel paths in the IODF are deleted and newly defined as they are defined in the IOCP input data set.

Example 4: Replacing a control unit that attaches to two processors

In this example, you want to replace existing control units and devices by another type. The control units and devices are attached to two processors.

The incremental update must be done in several steps:

1. Specify another control unit number in your IOCP input data set than the one in the existing IODF. Specify the CHPID, CNTLUNIT, and IODEVICE control statements.
2. Migrate your input data set for the first processor.

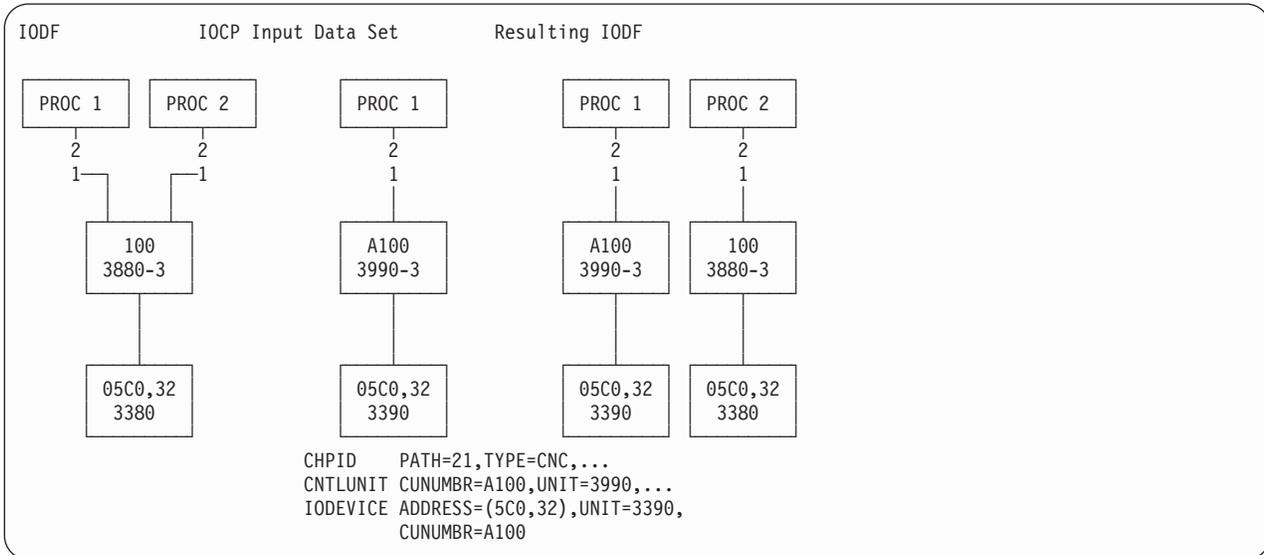


Figure 139. Partial migration of an IOCP input data set. Control unit defined for two processors is migrated to the first processor.

3. Migrate your input data set again for the second processor.

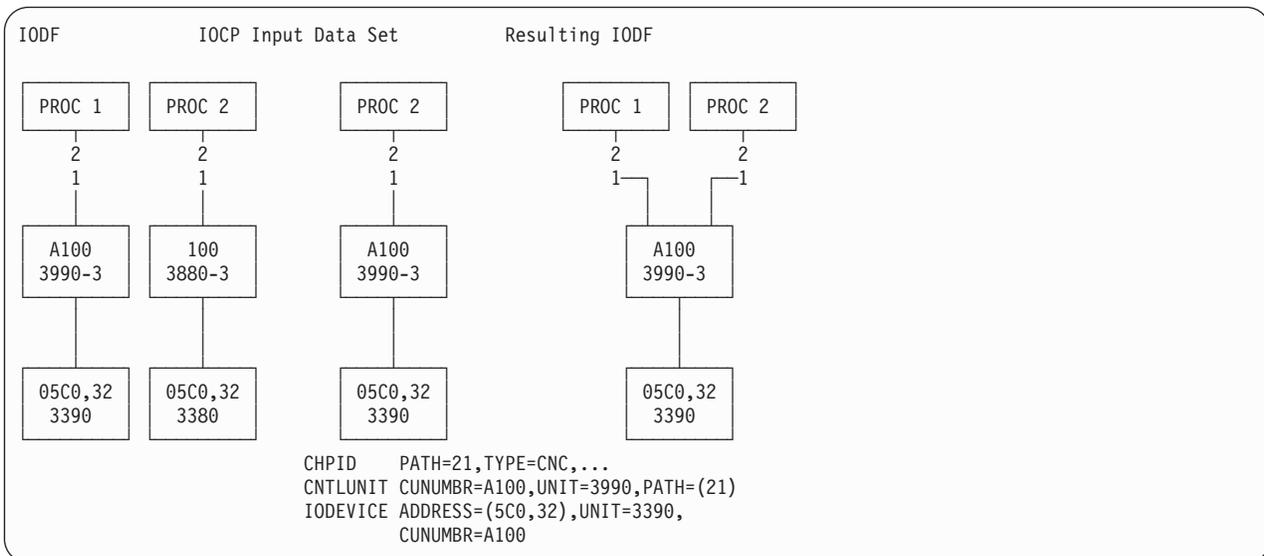


Figure 140. Partial migration of an IOCP input data set. Control unit defined for two processors is migrated to the second processor.

Example 5: Updating an operating system

In this example, you update an operating system by adding and changing EDTs, generics, esoterics, and console devices.

IODF	MVSCP Input Data Set
EDT ID=01 UNITNAME=POOL1,UNIT=((700,8),(900,8)) UNITNAME=POOL2,UNIT=((100,16)) UNITNAME=TAPES,UNIT=((A80,4))	IODEVICE ADDRESS=(800,8),UNIT=...,MODEL=... EDT ID=01 UNITNAME=POOL1,UNIT=((800,8)),VIO=YES UNITNAME=POOL3,UNIT=((0270,16)) UNITNAME=9345,VIO=YES
NIPCON DEVNUM=(40,41,42)	EDT ID=02 UNITNAME=TAPES,UNIT=((180,4)) NIPCON DEVNUM=(40,50,60)
<hr/>	
Resulting IODF	
IODEVICE ADDRESS=(800,8),UNIT=...,MODEL=... EDT ID=01 UNITNAME=POOL1,UNIT=((700,8),(800,8),(900,8)),VIO=YES UNITNAME=POOL2,UNIT=((100,16)) UNITNAME=POOL3,UNIT=((0270,16)) UNITNAME=TAPES,UNIT=((A80,4)) UNITNAME=9345,VIO=YES	
EDT ID=02 UNITNAME=TAPES,UNIT=((180,4))	
NIPCON DEVNUM=(040,050,060)	

Figure 141. Partial migration of an MVSCP input data set

You can either migrate the changes with a combined input data set or with an MVSCP only input data set. With a combined input data set, you can make CSS and operating system changes at the same time.

If you migrate an MVSCP only input data set, specify a processor and partition, with which device definitions in the input data sets will be associated in the IODF, on the Migrate IOCP / MVSCP / HCPRIO Data panel. HCD uses this information to map devices correctly if the IODF contains duplicate device numbers. For more information about this mapping, refer to “Migrating additional MVSCP or HCPRIO input data sets” on page 285.

The result after the migration is as follows:

- The definition of EDT input statements replaces the definition in the IODF, new definitions are added. That means in the example Figure 141, EDT 01 is replaced and EDT 02 is added.
- Devices are added. To add devices, you also have to newly define the esoterics to which the devices are to be assigned.
- Devices in a UNITNAME statement are added to the devices already defined for the esoteric.
- The list of consoles is replaced.

Note: If an EDT statement is missing or specified without EDT ID, the EDT ID for esoterics and generics is taken from the ID of the IOCONFIG statement. If the IOCONFIG statement is also not available, the EDT ID will be assumed as '00'.

Resolving migration errors

If HCD detects an error when you migrate using the HCD dialog, it displays a message list when the migration has been completed.

Figure 142 is an example of such a message list. It lists all the messages that were issued during the process. The list shows the statement number of the input statement for which the message was issued. You can now:

- Get explanations of messages in the list. To get explanations, use the *Explain message* action from the context menu or action code **e**.
- Delete messages that are of no further interest. That is, delete those messages that you do not want to appear in the *HCD migration log*.

```
Message List
-----
Query Help
-----
Row 1 of 6

Select one or more messages, then press Enter.

/ Statement Orig Sev Message Text
- 9         1   E Duplicate unit address F0 on channel path 01 of
#           processor B0EHCD.
- 5         2   W Type of control unit 0131 assumed as 3800-13 to
#           attach device 01F0.
#           I No output written to IODF. VALIDATE processing
#           forced due to errors.
```

Figure 142. Message List

When you exit the message list or when you migrated your input data sets using the batch utility, HCD writes the error messages to the HCD migration log.

You can display the HCD migration log through ISPF. The name of the HCD migration log data set is developed from the name of the input data set, without high-level qualifier, and your user ID as follows:

`userid.yyy.zzz.MESSAGES`

where the input data set, without high-level qualifier, is:

`yyy.zzz`

or

`yyy(zzz)`

Errors detected during assembly process

During migration, HCD invokes the assembler to parse the input statements. If it encounters an error, it writes a message to the message list. This message points to an assembler listing for more details.

```

Query Help
-----
Message List                               Row 1 of 2

Select one or more messages, then press Enter.

/ Statement Orig Sev Message Text
-           T   Assembler processing returns with return code   =
#           8. See BPAN.CDS.CSYSIO.LISTING for details.

```

Figure 143. Message List containing an Assembler message

The name of the assembly listing data set is developed from the name of the input data set, without high-level qualifier, and your user ID as follows:

`userid.yyy.zzz.LISTING`

where the input data set, without high-level qualifier, is:

`yyy.zzz`

or

`yyy(zzz)`

At the bottom of the assembly listing, you find the statement numbers where errors were detected. If you locate the statement numbers in the listing, you see the IOCP statement in error followed by the error message. The following examples show IOCP statements and the assembly messages issued.

Example 1

In this example an entry in the IOCP input data set is commented out, but the continuation character \$ is left in column 72.

```

7493 *          CNTLUNIT CUNUMBR=02E,PATH=(02,06),SHARED=N,UNIT=3990,      91A$
7494 *          UNITADD=((E),32)),PROTOCL=S4

```

The assembler listing shows the following message:

```
IEV144 *** ERROR *** BEGIN-TO-CONTINUE COLUMNS NOT BLANK
```

Example 2

This example shows a line with only 87A being included in the IOCP input data set.

```

7493 *          CNTLUNIT CUNUMBR=234,PATH=24,SHARED=N,UNIT=3880,          87A
7494 *          UNITADD=(30,16),PROTOCL=S4                                87A
7495                                                                 87A
7496 *IOCP                                                                87A

```

The assembler listing shows the following message:

```
IEV144 *** ERROR *** OPERATION CODE NOT COMPLETE ON FIRST CARD
```

Example 3

This example shows a wrong channel path type in the CHPID statement.

```
CHPID PATH=((00)),TYPE=CMC
```

The assembler listing shows the following message:

```
IEV144 *** MNOTE *** 8,003 TYPE=CMC IS INVALID
```

Errors detected during validation process

During the validation process, HCD checks that the definitions being migrated do not cause any conflicts with existing definitions in the IODF and with other definitions being migrated. HCD also checks that the contents of the input data sets is valid. The errors are shown in the message list and migration log.

Figure 144 shows error messages in the migration log.

```
I/O Configuration Migration          Time: 11.39   Date: 94-11-26   Page:   1
MVS/ESA HCD V 5.2

(1) DSN=BBEI.IOCP01.CTL      1
(2) DSN=BBEI.MVSCP01.CTL

Statement Orig Sev  Msgid  Message Text
-----
   9   (1)  E  CBDA230I Duplicate unit address F0 on channel path 01 of processor BOEHCD.
   5   (2)  W  CBDA265I Type of control unit 0131 assumed as 3880-13 to attach device 01F0.
        I  CBDA516I No output written to IODF. VALIDATE processing forced due to errors.

Total Messages  Terminating  Error  Warning  Informational
         3             0         1         1         1

Return Code was 8
```

Figure 144. Example: HCD Migration Log

The messages are sorted according to their severity, and within a certain severity level according to their occurrence.

The value in the **Orig** column points to the input data set that caused this error. At the start of the migration log you find a reference list that shows the values with the names of the input data sets (see the line marked **1**).

In Figure 144 the first message line means, that the statement number 9 in the input data set 1 (data set BBEI.IOCP01.CTL) is the cause of the error message.

The following examples show common validation errors and explain their causes.

Example 1

```
Statement Orig Sev  Msgid  Message Text
-----
   7   (1)  E  CBDA154I Channel path type CNC is not supported by channel
        path ID 3A.
```

This message is issued, because an ESCON channel is defined although the support level was defined in the IODF as having only parallel channels installed for the specified CHPID. To resolve this, either change the channel type in the IOCP input data set, or change the processor type or support level in the IODF.

Example 2

```
Statement Orig Sev  Msgid  Message Text
-----
   4   (1)  E  CBDA234I Unknown type 38823 of control unit 0000
        specified.
```

This message is issued, because HCD does not know the control unit type 38823. Select the *Query supported hardware and installed UIMs* from the *Primary Task Selection* panel or use the *Query* action bar choice for information on valid control unit and device types.

Example 3

Statement	Orig	Sev	Msgid	Message Text
228	(1)	W	CBDA265I	Type 3800-3 assumed for control unit DD32 to attach the device 0828.
227	(1)	I	CBDA534I	Control unit DD32 is assumed as 3800-1.

These messages are issued, because HCD has to choose the control unit type among several models.

The control unit model 3800-1 is indicated as default model in the UIM (information message CBDA534I is issued). As processing goes on, it is necessary to change the default model (3800-1) to another model (3800-3) to attach a device type which is not supported by the default control unit model (warning message CBDA265I is issued).

Note: The sequence of messages is shown in reverse order in the migration log file since the messages are sorted according to decreasing severities. To resolve this, either include the model in the IOCP input data set, map the control unit types via HCD profile entries, or update the IODF using the HCD dialog, if HCD has made an incorrect assumption.

Insufficient data set sizes

HCD dynamically allocates the data sets required for migration. It can happen that the default data set sizes are insufficient for the migration of the existing data. During the migration process, the system informs you by a message which data set needs to be enlarged. In that case, the data set needs to be preallocated with a larger size before invoking the HCD migration task again.

HCD uses, by default, the following ddnames and data set sizes:

- HCDPRINT

Used for the HCD migration log.

The data set name is built from the input data set name — if two input data sets are specified from the IOCP input data set name — qualified with 'MESSAGES'. If the input is a member of a partitioned data set, an additional qualifier - the name of the member - is inserted before 'MESSAGES'. The high-level qualifier of that data set name is replaced by the TSO prefix (user ID).

The minimum allocation (also used as default allocation if the data set does not exist) is: RECFM=FBA, LRECL=133, BLKSIZE=2926, SPACE=(TRK,(1,10)), exclusive access.

- HCDASMP

Contains the assembler SYSPRINT data set, which contains the assembly listing (input statements with sequence numbers and messages).

The data set name is built from the input data set name qualified with 'LISTING'. If the input is a member of a partitioned data set, an additional qualifier - the name of the member - is inserted before 'LISTING'. The high-level qualifier of that data set name is replaced by the TSO prefix (user ID).

The minimum allocation (also used as default allocation if the data set does not exist) is as follows: RECFM=FBA, LRECL=121, BLKSIZE=1573, SPACE=(TRK,(15,150)), exclusive access.

- HCDUT1

Used by the assembler as a work data set (UNIT=SYSALLDA, LRECL=80, BLKSIZE=3200, SPACE=(TRK,(15,150)), exclusive access).

- HCDUT2

Used as output data set for the modified input stream and by the assembler as SYSIN data set (UNIT=SYSALLDA, LRECL=80, SPACE=(TRK,(10,15)), exclusive access).

- HCDUT3

Used by the assembler as punch data set (SYSPUNCH).

Used by the loader as input data set (UNIT=SYSALLDA, LRECL=80, SPACE=(TRK,(10,30)), exclusive access).

Note:

1. The sizes of the output data sets HCDUT2, HCDASMP, and HCDPRINT depend on the size of the input and on the number of messages produced. The space given under "Insufficient data set sizes" on page 325 should be taken as minimum allocation values.
2. For HCDASMP and HCDPRINT, HCD checks whether data sets with the default names exist. If so, the space allocations of these existing data sets are used if they exceed the minimum allocation values. If they are below the minimum allocation value, the data sets are deleted and allocated with a new (minimum) size. Correspondingly, the space of the HCDUT2 data set is made dependent on the input data set(s). This rule is only applicable if the ddnames have not been previously allocated.
3. Preallocate HCDASMP and HCDPRINT if:
 - You want to have them on a different data set than the default one
 - You want to place these data sets on a specific volume
 - The default size is not large enough
 - A size different from the default size should be used.

Chapter 13. How to invoke HCD batch utility functions

Programming Interface Information

You can invoke HCD batch utility functions:

- With an input parameter string. The following syntax diagram shows how to invoke an HCD batch utility function. For formats of the input parameter strings and sample batch jobs, see “Input parameter string” on page 328.

HCD Batch Invocation

```
▶▶ EXEC PGM=CBDMGHCP, PARM=' | Input Parameter String | '▶▶
```

- By using an ATTACH or LINK module programming statement to invoke the module CBDMGHCP.

When you invoke the module, register 1 must contain the address of a two-word parameter list.

Word 1

Address of input parameter (see “Input parameter string” on page 328), preceded by a two byte length field.

Word 2

Address of a list of alternate DD names. If not used, the address must be binary zero. For the list format of alternate DD names see “List of alternate DD names” on page 358.

You may overwrite standard DD names listed in Table 24 on page 357 as desired before invoking HCD.

If you specify the UIMs and UDTs in a library other than SYS1.NUCLEUS, you have to add the following statement to your batch jobs:

```
//HCDPROF DD DSN=BPAN.HCD.PROF,DISP=SHR
```

In the HCD profile (in our example BPAN.HCD.PROF) specify the following keyword:

```
UIM_LIBNAME=i bname
```

If the keyword is omitted, SYS1.NUCLEUS is assumed. If you specify an asterisk (*) as data set name, HCD assumes that the UIM data set is part of the ISPF load library concatenation chain, contained in the JOBLIB/STEPLIB concatenation chain, or specified in the active LNKLISTxx member. For more information, see “Defining an HCD profile” on page 19.

Running jobs in a sysplex environment:

If you want to execute a job on a specific system in a sysplex, you must specify in your batch job which system is to be used. If you do not specify the exact system of a sysplex for which the batch job is planned to execute on, the job executes on the system that has the free space to run on.

The output from some of the HCD functions depend very much on where the job was executed; for example, downloading IOCDs and requesting the I/O Path report.

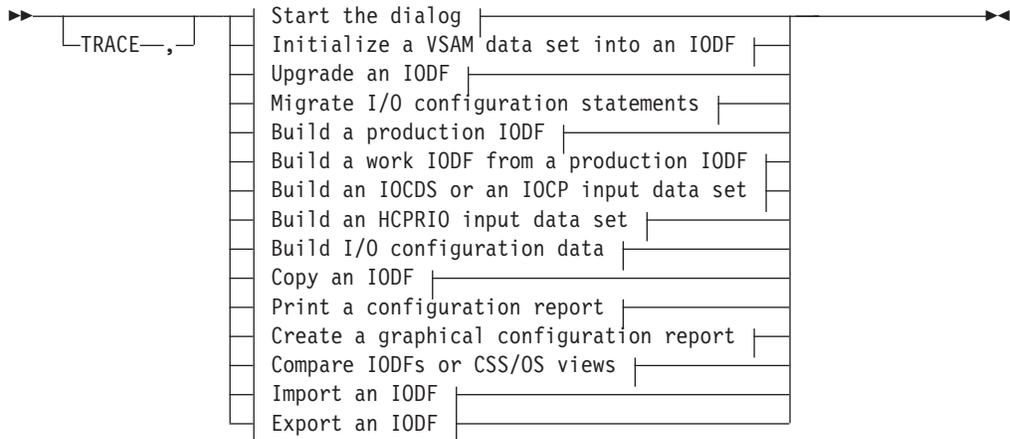
How to Read Syntax Diagrams

For details on this subject see “How to read syntax diagrams” on page xviii.

Note: Trailing commas in the parameter string can be omitted.

Input parameter string

Input Parameter String



TRACE

When specified, the HCD trace will be activated.

Input Parameter String

You will find a detailed description of the input parameter strings in the following sections.

You can also activate tracing by adding the TRACE command in the HCD profile. This allows you to specify the trace parameters in more detail. In this case, you must allocate DD name HCDPROF to the HCD profile when invoking the batch utility. For more information, see “Defining an HCD profile” on page 19, and “TRACE command” on page 491.

Start the dialog

This utility function starts the HCD (ISPF) Dialog session. The HCD primary task selection panel is displayed.

This function is invoked by passing the following parameter string.

Start the dialog



E | J

is a one-character code for national language support in help panels and messages. Specify one of the following (if omitted, the default is E):

E for English

Note:

1. This batch job issues a job message IEC161I, which can be ignored.
2. The VSAM DIV file consists of a data and a cluster file. According to the IODF naming convention (see "IODF naming convention" on page 31), the name of the data file is the IODF name (in this example SYS1.IODF01), and .CLUSTER is appended to the data file for a cluster file. To define your VSAM DIV file, you must append .CLUSTER to the IODF name in the DEFINE CLUSTER statement (in this example SYS1.IODF01.CLUSTER).

Upgrade IODF

This utility function upgrades a back-level IODF to be accessible with the current HCD release.

You have to allocate:

- The IODF you want to upgrade with DD name HCDIODFS
- The IODF into which the I/O definitions are to be upgraded with DD name HCDIODFT

This function is invoked by passing the following parameter string.

Upgrade an IODF

UPGRADE

Note:

1. The target data set must be large enough to hold the source IODF.
2. The target IODF may be created using the utility *Initialize IODF* (described in "Initialize IODF" on page 329).
3. Both data sets must be valid IODFs.

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
HCDIODFS	Back-level IODF to be upgraded
HCDIODFT	IODF into which IODF definitions are to be upgraded (if not specified, the IODF is upgraded in place)
HCDMLOG	HCD Message Log data set
HCDTRACE	Trace data set (if trace is activated)

Example:

```
//BWINJOB JOB (3259,RZ-28), 'BWIN', NOTIFY=BWIN, CLASS=A,
//          MSGCLASS=Q, MSGLEVEL=(1,1), REGION=4M
// *
// * UPGRADE IODF
// *
//UPGRADE EXEC PGM=CBDMGHCP, PARM='UPGRADE'
//HCDIODFS DD DSN=BWIN.IODFR2.WORK, DISP=SHR
//HCDIODFT DD DSN=BWIN.IODF00.WORK, DISP=OLD
//HCDMLOG DD DSN=BWIN.HCD.LOG, DISP=OLD
//
```

For considerations concerning the size when upgrading a back-level IODF, refer to Table 3 on page 45.

Migrate I/O configuration statements

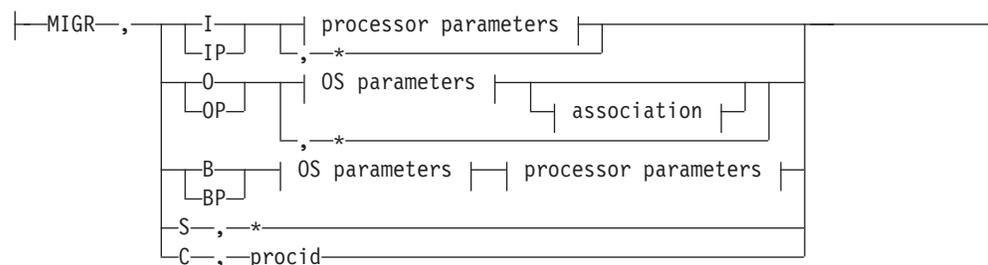
This utility function allows you to migrate the data set containing I/O configuration statements, e. g. an IOCP, MVSCP, or HCPRIO input data set and store the definitions into an IODF.

You have to allocate:

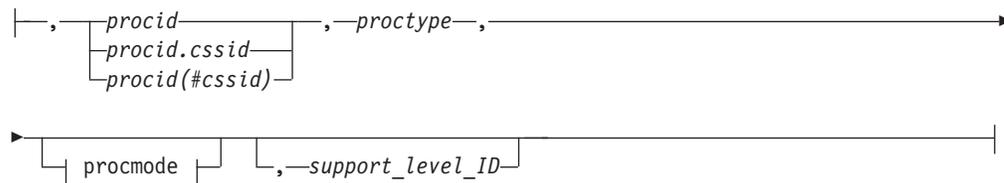
- The IODF into which the I/O definitions are to be migrated with DD name HCDIODFT
- The I/O configuration input data set with DD name HCDIN
- The MACLIB containing the parsing macros with DD name HCDLIB

This function is invoked by passing the following parameter string.

Migrate I/O configuration statements



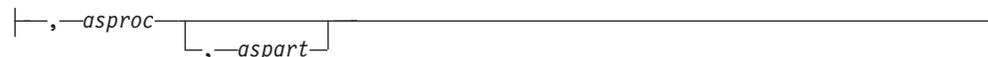
processor parameters:



OS parameters:



association:



procmode:



I | IP | O | OP | B | BP | S | C

Function indicator:

- I** Migration of processor configuration statements (e. g. IOCP data sets)
- IP** Partial migration of processor configuration statements
- O** Migration of OS configuration statements (for example MVSCP or HCPRIO data sets)
- OP** Partial migration of OS configuration statements
- B** Combined migration of processor and OS configuration statements
- BP** Partial combined migration of processor and OS configuration statements
- S** Switch migration
- C** physical channel ID (PCHID) migration

Processor related variables and keywords:

procid Processor ID (up to 8 characters)

procid.cssid, procid(#cssid)

When migrating an SMP processor to an XMP processor, the channel subsystem ID of the target processor may be appended to the processor ID as one character either by a # and in parenthesis or by a dot (.). The default is 0.

proctype

Processor type and model separated by a hyphen, for example 9672-E08

procmode

Processor mode

BASIC

If the processor operates in BASIC mode (default).

LPAR If the processor operates in LPAR mode.

support_level_ID

Support level ID associated with the processor. This parameter is required if the processor does not already exist and several support level IDs are installed for a supported processor type. The support level ID can be obtained by the *List supported processors* function or by the supported hardware report described in "Print configuration reports" on page 343. For an example of a supported hardware report refer to "Supported hardware report" on page 425.

If you do not specify a support level, the highest support level will be used for the processor.

Note: The support level ID is unique to HCD and does not correspond to the EC level of the processor.

- * Allows multiple processor configurations to be migrated. Scans the input data set to determine which processor configurations are to be processed. For successful migration the configurations must include the ID statement described in "Processor" on page 300.

OS related parameters:

- osid** Operating system ID (up to 8 characters)
- ostype** OS type (MVS or VM)
- asproc** Associated processor. For more information, see “Migrating additional MVSCP or HCPRIO input data sets” on page 285
- aspart** Associated partition. For more information, see “Migrating additional MVSCP or HCPRIO input data sets” on page 285
- * Allows multiple OS configurations to be migrated. Scans the input data set to determine which OS configurations are to be processed. For successful migration the configurations must include the IOCONFIG statement described in “Operating system” on page 294.

Switch related wildcard:

- * Allows switch configurations of multiple switches to be migrated. Scans the input data set to determine which switch configurations are to be processed. For successful migration the configurations must include the SWCONF statement described in “Switch configuration” on page 299.

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
HCDIODFT	IODF into which I/O definitions are to be migrated
HCDIN	I/O configuration input data set
HCDLIB	MACLIB containing the parsing macros (CBDZPARS)
HCDMLOG	HCD Message Log data set
HCDPRINT	Data set for migration log (see “Insufficient data set sizes” on page 325)
HCDASMP	Data set for assembly listing (see “Insufficient data set sizes” on page 325)
HCDPROF	HCD profile (when using extended migrate function)
HCDTRACE	Trace data set (if trace is activated)

For defaults of HCDPRINT and HCDASMP, for preallocating additional migration data sets, and for viewing the migration log see “Resolving migration errors” on page 322. **Example:**

```
//BWINJOB JOB (3259,RZ-28), 'BWIN', NOTIFY=BWIN, CLASS=A,
//          MSGCLASS=Q, MSGLEVEL=(1,1), REGION=4M
// *
// * MIGRATE AN IOCP DECK
// *
//MIGRATE EXEC PGM=CBDMGHCP,
//          PARM='MIGR,I,PROC1,9672-E08,LPAR'
//HCDIODFT DD DSN=BWIN.IODF03.WORK, DISP=OLD
//HCDIN DD DSN=BWIN.IOCP.DECK, DISP=SHR
//HCDLIB DD DSN=SYS1.MACLIB, DISP=SHR
//HCDMLOG DD DSN=BWIN.HCD.LOG, DISP=OLD
//HCDPRINT DD DSN=BWIN.IOCP.MESSAGES, DISP=OLD
//HCDASMP DD DSN=BWIN.IOCP.LISTING, DISP=OLD
//
```

Build a Production IODF

This utility function creates a production IODF using the work IODF. The work IODF has to be specified with DD name HCDIODFS, the target production IODF with DD name HCDIODFT. First, the target production IODF has to be created by defining a VSAM DIV file and initializing it using the utility Initialize IODF (see "Initialize IODF" on page 329). If the target IODF is the active IODF (used for IPL and holding the active processor configuration) the IODF will not be replaced and build production IODF will be rejected.

If the work IODF has an associated MCF, the MCF data set is copied and associated to the production IODF.

After the production IODF has been built, it is copied back to the work IODF. Thus, the work IODF contains the new tokens and can be used for further updates. Since the production IODF may be larger than the original work IODF, the work IODF may be automatically enlarged to accommodate the contents of the production IODF.

This function is invoked by passing the following parameter string.

Build a Production IODF

```

|-----PRODIODF-----|
|-----DESC1==--descriptor1-----|
|-----DESC2==--descriptor2-----|
|-----DESC1==--descriptor1--,--DESC2==--descriptor2-----|
  
```

DESC1=*descriptor 1*

Default is the first qualifier of the production IODF name (up to 8 characters).

DESC2=*descriptor 2*

Default is the second qualifier of the production IODF name, which is IODFxx (up to 8 characters).

The descriptor fields describe the IODF and will be part of the HSA token.

Attention: If you specify asterisks (**), equals (==), pluses (++), or minuses (--) for the IODF suffix in LOADxx, never change the default descriptor field values, because z/OS uses these values to find the current IODF during IPL. Take this relationship also into consideration, if you copy the IODF to a different data set name.

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
HCDIODFS	Work IODF
HCDIODFT	Production IODF
HCDMLOG	HCD Message Log data set
HCDTRACE	Trace data set (if trace is activated)

Example:

```

//BWINJOB JOB (3259,RZ-28),'BWIN',NOTIFY=BWIN,CLASS=A,
//          MSGCLASS=Q,MSGLEVEL=(1,1),REGION=4M
//*
//* BUILD PRODUCTION IODF
//*
//PROD      EXEC PGM=CBDMGHCP,
//          PARM='PRODIODF DESC1=BWIN,DESC2=IODF03'
//HCDIODFS DD DSN=BWIN.IODF03.WORK,DISP=OLD
//HCDIODFT DD DSN=BWIN.IODF03,DISP=OLD
//HCDMLOG  DD DSN=BWIN.HCD.LOG,DISP=OLD
//

```

Build a work IODF from a production IODF

This utility function creates a work IODF using an existing production IODF. The production IODF has to be specified with DD name HCDIODFS, the target work IODF with DD name HCDIODFT. First, the work IODF has to be created by defining a VSAM DIV file and initializing it using the utility Initialize IODF (see “Initialize IODF” on page 329). If the target IODF is the active IODF (used for IPL and holding the active processor configuration) the IODF will not be replaced and build work IODF will be rejected.

This function is invoked by passing the following parameter string.

Build a Work IODF

|—WORKIODF—|

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
HCDIODFS	Production IODF
HCDIODFT	Work IODF
HCDMLOG	HCD Message Log data set
HCDTRACE	Trace data set (if trace is activated)

Example:

```

//BWINJOB JOB (3259,RZ-28),'BWIN',NOTIFY=BWIN,CLASS=A,
//          MSGCLASS=Q,MSGLEVEL=(1,1),REGION=4M
//*
//* BUILD WORK IODF
//*
//WORK      EXEC PGM=CBDMGHCP,PARM='WORKIODF'
//HCDIODFS DD DSN=BWIN.IODF03,DISP=SHR
//HCDIODFT DD DSN=BWIN.IODF03.WORK,DISP=OLD
//HCDMLOG  DD DSN=BWIN.HCD.LOG,DISP=OLD
//

```

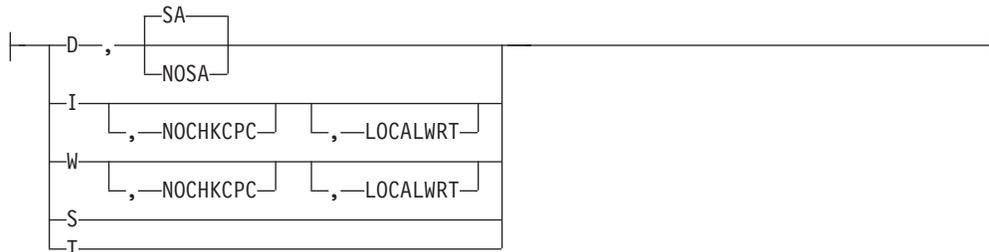
Build an IOCDS or an IOCP input data set

This utility function builds the IOCDS or the IOCP input data set using the definitions of a production IODF. This function is invoked by passing the following parameter string.

Build an IOCP or an IOCP input data set

► IOCDs, iocds, *procid*, | ind |

ind



iocds

is a two-character IOCDs identifier, if building an IOCDs

procid

Processor ID

D | I | W | S | T

One-character request code:

D Build an IOCP input data set

SA The generated IOCP statements can be used for the stand-alone IOCP program (default).

NOSA

Depending on the HCD profile option MIGRATE_EXTENDED=YES, the generated IOCP statements have additional information that can be used for the extended migration. This information is shown as comments to IOCP.

Note: An IOCP input data set generated with operand NOSA may not be accepted by the stand-alone IOCP program, because of differences between the IOCP program running in z/OS and the stand-alone IOCP program.

I Build an IOCDs

NOCHKCPC

Write an IOCDs regardless of the type of the receiving processor. Refer to "Supported hardware report" on page 425 for a list of processor types that can receive an IOCDs in preparation for a processor upgrade and for processor types for which such an IOCDs can be written.

LOCALWRT

This parameter enforces a local IOCDs write. A defined SNA address which is normally used by HCD to initiate a remote IOCDs build to the support element with the designed SNA address, is ignored in this case.

W Build an IOCDs with dual-write option (optionally with **NOCHKCPC** and **LOCALWRT**, see option I).

- S Build an IOCDS and set the IOCDS active for next POR
- T Build an IOCDS with dual-write option and set the IOCDS active for next POR

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
HCDIODFS	Source IODF
HCDDECK	IOCP data set (if build IOCP input data set is requested)
HDCDNTL	Control data set for specifying the MSG1 IOCP parameter
SYSPRINT	SYSPRINT data set for IOCP output listing (requested for build IOCDS)
SYSIN	Temporary work file used as IOCP input deck
HCDMLOG	HCD Message Log data set
HCDPROF	HCD profile (when generating additional information for extended migration)
HCDTRACE	Trace data set (if trace is activated)

Example 1:

The following example shows a *Build IOCDS* job.

```
//BWINJOB JOB (3259,RZ-28),'BWIN',NOTIFY=BWIN,CLASS=A,
//          MSGCLASS=Q,MSGLEVEL=(1,1),REGION=5M
//*
/* BUILD IOCDS (WITH DUAL-WRITE OPTION)
/*
//IOCDS EXEC PGM=CBDMGHCP,
//          PARM='IOCDS,A0,PROC1,W'
//HCDIODFS DD DSN=BWIN.IODFA3,DISP=OLD
//HCDMLOG DD DSN=BWIN.HCD.LOG,DISP=OLD
//SYSPRINT DD DSN=&&IOCPOUT,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6650),
//          SPACE=(CYL,(1,1)),DISP=(NEW,PASS,DELETE),UNIT=SYSALLDA
//SYSIN DD DSN=&&TEMP,DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),UNIT=SYSALLDA
//HDCDNTL DD *
IOCDSNAM
/*
/* EXECUTED ONLY IF RETURN CODE HIGHER THAN 0
//PRINT EXEC PGM=IEBGENER,COND=(0,EQ,IOCDS)
//SYSUT1 DD DSN=&&IOCPOUT,DISP=(OLD,DELETE)
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD DUMMY
/*
//CLEANUP EXEC PGM=IEFBR14,COND=(0,NE,IOCDS)
//SYSUT DD DSN=&&IOCPOUT,DISP=(OLD,DELETE)
//
```

Figure 145. Build IOCDS job

Example 2:

The following example shows a *Build IOCP input data set* job.

```

//BWINJOB JOB (3259,RZ-28),'BWIN',NOTIFY=BWIN,CLASS=A,
//          MSGCLASS=Q,MSGLEVEL=(1,1),REGION=5M
//*
//* BUILD IOCP DECK
//*
//IOCP      EXEC PGM=CBDMGHCP,
//          PARM='IOCDS,,PROC1,D,SA'
//HCDIODFS DD DSN=BWIN.IODFA3,DISP=SHR
//HCDDECK  DD DSN=BWIN.IOCP3.DECK,DISP=OLD
//HCDMLOG  DD DSN=BWIN.HCD.LOG,DISP=OLD
//HDCNTL   DD *
IOCDSNAM
/*
//

```

Figure 146. Build IOCP input data set

Note: HDCNTL specifies the value of the MSG1 parameter (in the example: IOCDSNAM) which is the identification information printed on the first ID line of the heading of the IOCP input data set. Specify the text without any keyword and quotation-marks. The first eight characters are used as IOCDS name.

Activate a production IODF

This utility function activates an I/O configuration from an existing production IODF. Both the active IODF and the target IODF have to be accessible.

This function is invoked by passing the ACTIVATE command in the parameter string (PARM='...'), as shown in the example. The parameter string uses the same syntax as described for the ACTIVATE command in *z/OS MVS System Commands*.

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
HCDMLOG	HCD Message Log data set
HCDTRACE	Trace data set (if trace activated)

Example

```

//BBEIJOB JOB (3259,RZ-28),'BBEI',NOTIFY=BBEI,CLASS=A,
//          MSGCLASS=Q,MSGLEVEL=(1,1),REGION=0M
//*
//* ACTIVATE PRODUCTION IODF
//*
//WORK      EXEC PGM=CBDMGHCP,PARM='ACTIVATE IODF=01,TEST'
//HCDMLOG  DD DSN=BBEI.HCD.MSGLOG,DISP=OLD
//

```

Figure 147. Activate IODF job

Build an HCPRIO input data set

This utility function creates an HCPRIO input data set using the definitions of a VM operating system in a production IODF.

The IODF from which I/O definitions are extracted has to be specified with DD name HCDIODFS, and the HCPRIO input data set with DD name HCDDECK.

This function is invoked by passing the following parameter string.

Build an HCPRIO input data set

|—VMBUILD—,—H—,—*osid*—|

H Function indicator for HCPRIO input data set

osid

VM operating system ID (up to 8 characters)

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
HCDIODFS	IODF from which I/O definitions are extracted
HCDDECK	VM I/O configuration data set
HCDMLOG	HCD Message Log data set
HCDTRACE	Trace data set (if trace is activated)

Example:

```
//BWINJOB JOB (3259,RZ-28), 'BWIN', NOTIFY=BWIN, CLASS=A,
//          MSGCLASS=Q, MSGLEVEL=(1,1), REGION=4M
// *
// * BUILD AN HCPRIO INPUT DATA SET
// *
//HCPRIO EXEC PGM=CBDMGHCP, PARM='VMBUILD,H,VM1'
//HCDIODFS DD DSN=BWIN.IODF03, DISP=SHR
//HCDDECK DD DSN=BWIN.HCPRIO3.DECK, DISP=OLD
//HCDMLOG DD DSN=BWIN.HCD.LOG, DISP=OLD
//
```

Build I/O configuration data

This utility function allows you to build I/O configuration statements from an IODF and to store them in a data set. The statements describe:

- Operating system configurations
- Processor configurations
- Switch configurations

The data sets created can be edited and re-migrated into the IODF.

In addition, you can build:

- JES3 initialization stream checker data
- CONFIGxx members (from production IODF only)
- FCP device data

You invoke this function by passing the following parameter string:

Build I/O configuration data

- FCP** Export FCP device data into CSV output format
- procid** Processor ID (up to 8 characters) for which to export the FCP device data.
- JES** Build JES3 initialization stream checker data
- osid** OS configuration ID (up to 8 characters)
- edt_id** EDT ID (2 hexadecimal characters)
- XX** Build CONFIGxx members
- xx** Suffix of the CONFIGxx member to be built
- procid** Processor ID (up to 8 characters)
- partition_id**
Partition name; required entry, if the processor is in LPAR mode (up to 8 characters).
- osid** OS configuration ID (up to 8 characters)
- U/R** U updates the current CONFIGxx member. The CHP and DEVICE statements are replaced and all other statements remain unchanged. This is the default.
- R deletes the current CONFIGxx member and generates new CHP and DEVICE statements.
- backup**
Name for the backup copy of the current CONFIGxx member (up to 8 characters).
- sysplex**
Name of the sysplex used for setting managed channel paths to ONLINE.

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
HCDIODEFS	Source IODF
HCDDECK	Generated output data set For CONFIGxx this must be a data set name of a partitioned data set.
HCDMLOG	HCD Message Log data set
HDCNTL	Optional for specifying a list of operating systems, processors, or switches Not applicable for building CONFIGxx member or JES3 inish data set
HCDTRACE	Optional for capturing the trace if trace is activated.

Example 1:

The following example shows a job to build a configuration data set containing processor configuration PROC1 including its CF connections to processor PROC2.

```
//BWINJOB JOB (3259,RZ-28), 'BWIN', NOTIFY=BWIN, CLASS=A,
//          MSGCLASS=Q, MSGLEVEL=(1,1), REGION=4M
//*
```

```

/** BUILD Processor configuration statement
/**
//BUILD EXEC PGM=CBDMGHCP,PARM='CONFIG,PR,*'
//HCDIODFS DD DSN=BWIN.IODF03,DISP=SHR
//HCDDECK DD DSN=BWIN.IODF.03.DECKS(PROC1),DISP=SHR
//HCDMLOG DD DSN=BWIN.HCD.LOG,DISP=OLD
//HDCNTL DD *
PROC1
PROC2,CF
/*
//

```

Example 2:

The following example shows a job to update CONFIG03 in data set SYS1.PARMLIB from processor configuration PROC1, partition LPAR1 and OS configuration MVS1 while saving the existing member under the name CONFBK03.

```

//BWINJOB JOB (3259,RZ-28),'BWIN',NOTIFY=BWIN,CLASS=A,
//          MSGCLASS=Q,MSGLEVEL=(1,1),REGION=4M
/**
/** BUILD CONFIGxx
/**
//BUILD EXEC PGM=CBDMGHCP,PARM='CONFIG,XX,03,PROC1,LPAR1,MVS1,U,CONFBK03'
//HCDIODFS DD DSN=BWIN.IODF03,DISP=SHR
//HCDDECK DD DSN=SYS1.PARMLIB,DISP=SHR
//HCDMLOG DD DSN=BWIN.HCD.LOG,DISP=OLD
/*
//

```

Copy IODF

This utility function copies the content of the IODF, addressed by DD name HCDIODFS, into another data set, addressed by DD name HCDIODFT. If the IODF has an associated activity log, that log is also copied. Likewise, if an HCM master configuration file (MCF) is associated to the IODF, it is also copied along with the IODF. However, a change log file (CHLOG), if available, is not copied. If the target IODF is the active IODF (used for IPL and holding the active processor configuration) the IODF will not be replaced and copy IODF will be rejected.

This function is invoked by passing the following parameter string.

Copy an IODF

```

|-----COPYIODF-----|

```

Note:

1. The target data set must be large enough to hold the source IODF.
2. The target IODF can be created by defining a VSAM DIV file and by initializing it using the utility *Initialize IODF* (described in "Initialize IODF" on page 329).
3. Both data sets must be valid IODFs.
4. If you copy an IODF which is enabled for multi-user access, this property is not inherited by an existing target IODF. However, a target IODF defined with the multi-user access property will always preserve this property, independent from the source IODF.

- M OS report - OS devices
- E OS report - EDTs
- N OS report - NIP/VM consoles
- T CTC report
- I I/O Path report
- X Supported hardware report
- Y I/O definition reference

procid

Processor ID to limit a CSS, CTC connection or I/O Path report to a specific processor. If not specified for an I/O Path report, the ID of the active processor configuration is taken (=default).

partnm

Partition name to limit a CSS, CTC connection or I/O Path report to a specific logical partition. The processor ID must also be specified; otherwise, the partition name is ignored for the CSS and CTC connection report. For the I/O Path report, the partition name is defaulted to a partition that contains a device common to the specified or defaulted OS configuration.

osid

Operating system configuration ID to limit an OS report or an I/O Path report to a specific operating system configuration. If not specified for an I/O Path report, the ID of the active operating system configuration is taken (=default).

swid

Switch identifier to limit a Switch report to a specific switch

system

If the sysplex name is also specified, the system name (1 - 8 alphanumeric characters) identifies the system of a sysplex for that the I/O Path report is to be generated. If the sysplex is not specified, the system name is the VTAM application name of the host for that the I/O Path report is to be generated. The default is the local system.

sysplex

Sysplex name (1 - 8 alphanumeric characters) to specify the sysplex of the system for which the I/O Path report is to be generated. If the I/O path uses zDAC, the sysplex must be the local sysplex or left blank.

XML

Specify the XML keyword if you want to print your report in XML output format.

Note: It is recommended to print the I/O Path report separately from the other reports. However, if you want to print an I/O Path report together with any other reports, your limitations for the I/O Path report are propagated to all other specified reports. These limitations to a certain processor, partition or operating system can be either user-specified or HCD takes the active processor, partition or operating system as the default.

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
STEPLIB	SYS1.SCBDHENU (required for I/O definition reference)
HCDIODFS	Source IODF (not required when printing the supported hardware report, and I/O definition reference)
HCDRPT	Output data set: record size 133, record format fixed block
HCDMLOG	HCD Message Log data set
HCDTRACE	Trace data set (if trace is activated)

Example 1:

```
//BWINJOB JOB (3259,RZ-28), 'BWIN',NOTIFY=BWIN,CLASS=A,
//          MSGCLASS=Q,MSGLEVEL=(1,1),REGION=4M
//*****
//* PRINTS A CSS SUMMARY REPORT FOR PARTION PART1 OF PROCESSOR
//*          PROC1,
//*          A SWITCH REPORT FOR SWITCH 00,
//*          AN OS REPORT FOR DEVICES, EDT AND NIP CONSOLES OF
//*          OS CONFIGURATION MVS1
//*****
//REPORT1 EXEC PGM=CBDMGHCP,
//          PARM='REPORT,CSMEN,PROC1,PART1,MVS1,00'
//HCDIODFS DD DSN=BWIN.IODF03.WORK,DISP=SHR
//HCDRPT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6650)
//HCDMLOG DD DSN=BWIN.HCD.LOG,DISP=OLD
//
```

Example 2:

```
//BWINJOB JOB (3259,RZ-28), 'BWIN',NOTIFY=BWIN,CLASS=A,
//          MSGCLASS=Q,MSGLEVEL=(1,1),REGION=4M
//*****
//* PRINTS A SUPPORTED HARDWARE REPORT AND
//*          AN I/O DEFINITION REFERENCE
//*****
//REPORT2 EXEC PGM=CBDMGHCP,
//          PARM='REPORT,XY'
//STEPLIB DD DSN=SYS1.SCBDHENU,DISP=SHR
//HCDRPT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6650)
//HCDMLOG DD DSN=BWIN.HCD.LOG,DISP=OLD
//
```

Example 3:

```
//BWINJOB JOB (3259,RZ-28), 'BWIN',NOTIFY=BWIN,CLASS=A,
//          MSGCLASS=Q,MSGLEVEL=(1,1),REGION=4M
//*****
//* PRINTS AN I/O PATH REPORT OF THE ACTIVE CONFIGURATION
//*          COMPARED TO THE DEFINITIONS IN IODF SYS1.IODF00
//*****
//REPORT3 EXEC PGM=CBDMGHCP,
//          PARM='REPORT,I'
//HCDIODFS DD DSN=SYS1.IODF00,DISP=SHR
//HCDRPT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6650)
//HCDMLOG DD DSN=BWIN.HCD.LOG,DISP=OLD
//
```

Example 4:

```
//BHEIREP JOB (DE03243,,RZ-29), 'HEISSER',CLASS=A,REGION=4M,
//          MSGLEVEL=(1,1),NOTIFY=BHEI,MSGCLASS=Q
//*****
//**EXAMPLE OF A BATCH JOB THAT IS CREATED FROM THE DIALOG AND THAT
//**PRINTS AN I/O PATH REPORT OF THE ACTIVE CONFIGURATION OF THE
```


CUTYPE

limit the CU report by control unit type for supported types (see "Supported hardware report" on page 425).

CUGRP

limit the CU report by control unit group; valid values are: DASD, TAPE, CLUSTER, U/R, COMM, MICR/OCR, GRAPHIC, OTHER.

SWITCH

Switch report

CF CF connection report

CHPID

CHPID report

LCU LCU report

Procid Processor ID for which the LCU or CHPID report is produced.

Partname

Partition name to limit an LCU or CHPID report to one partition.

OPT Options of the report. Specify one or more of the following codes in any order, separated by a comma:

EP Exclude partition

ES Exclude switch

EC Exclude CTC control units

SD Show control unit description (instead of serial number)

EI Exclude index

DCF DCF output format

GML GML output format

Note:

1. If no output format is specified, the specification in the HCD profile is used. If the HCD profile does not specify a formatting type either, the default BookMaster format is used.
2. The output format GDF is not supported in batch mode.

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
HCDIODEFS	Source IODF
HCDRPT	Output data set: record size 200, record format FB. Note: This must be a cataloged data set.
HCDMLOG	HCD Message log data set
HCDPROF	HCD profile data set (if profile contains keywords concerning the graphical report)
HCDTRACE	Trace data set (if trace is activated)

Example:

```

//BWINGCR1 JOB (DE3259,,71034-83),'BWIN',NOTIFY=BWIN,CLASS=A,
//          MSGCLASS=Q,MSGLEVEL=(1,1),REGION=4M
//* -----
//* Graphical Configuration Report
//* -----
//GCREP EXEC PGM=CBDMGHCP,
//          PARM='GRAPHIC TYPE=CHPID,PROC=TEST3'
//HCDIODFS DD DSN=USER.IODF00.DBR4,DISP=SHR
//HCDRPT DD DSN=USER.IODF00.DBR4.REPORT,
//          DCB=(RECFM=FBA,LRECL=200,BLKSIZE=6400),
//          SPACE=(TRK,(50,50)),DISP=(NEW,KEEP),UNIT=SYSALLDA
//HCDMLOG DD DSN=USER.HCD.LOG,DISP=OLD
//HCDPROF DD DSN=USER.HCD.PROF,DISP=SHR
//

```

Compare IODFs or CSS/OS Reports

This utility function allows you to compare two IODFs and report the differences. You can compare the IODFs from the CSS, OS, and switch perspective.

In addition, you can limit the CSS, OS, and switch perspective by single compare reports, and the CSS perspective by LPARs.

You have to allocate the new IODF with DD name HCDIODFS, and the old IODF with DD name HCDIODFT for comparing IODFs. If you compare the CSS to the OS definition, you always compare within one IODF that must be allocated to HCDIODFS.

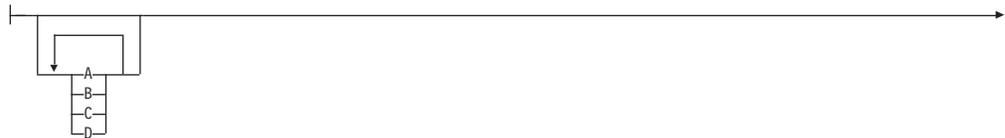
You invoke this function by passing the following parameter string.

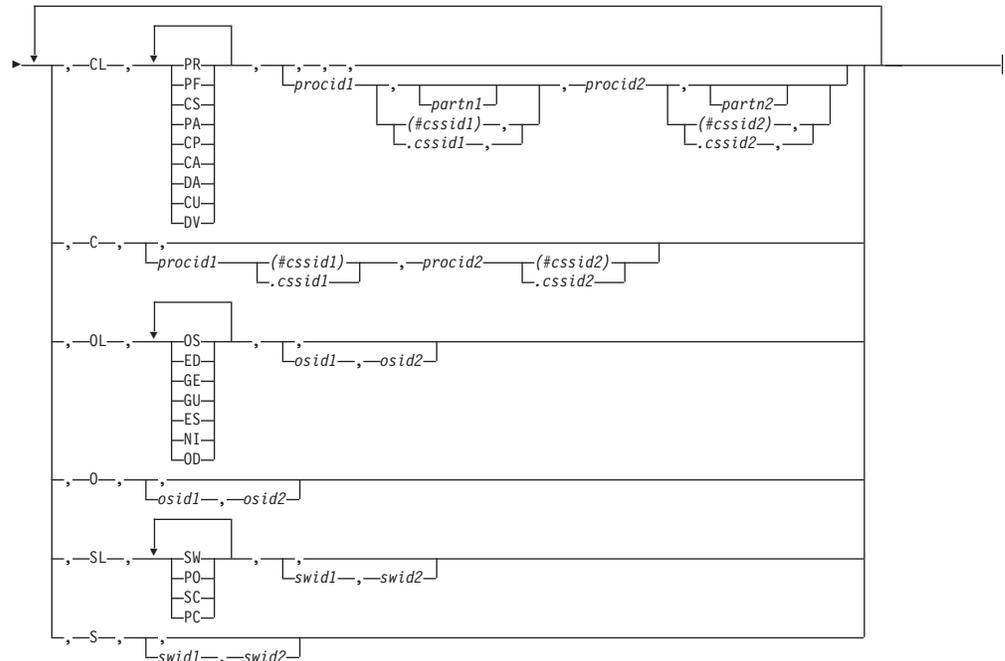
Note: This parameter string must not exceed 100 characters.

Compare IODFS or CSS/OS Reports

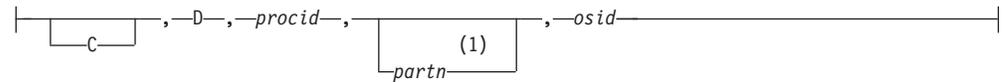
—COMPARE—, — Print Options for IODF Compare Report —
 — Print Options for CSS/OS Compare Report —

Print Options for IODF Compare Report:





Print Options for CSS/OS Compare Report:



Notes:

1 Required if processor runs in LPAR mode.

Print options for IODF Compare Report

Specify one or more of the following print options, without separating characters, in exactly this order. For more information about the print option types, see “How to print a Compare IODFs Report” on page 268.

- A** Print added data
- B** Print deleted data
- C** Print unchanged data
- D** Print unchanged item IDs

CL | C | OL | O | SL | S

Type of the report. Specify one or more of the following codes in any order, with no separating characters:

CL Limit the CSS compare report by single compare reports and LPARs

- PR** Processor compare
- PF** PCIe Function compare
- CS** Channel subsystem compare
- PA** Partition compare

- CP Channel path compare
- CA Control unit attachment compare
- DA Device attachment compare
- CU Control unit compare
- DV Device compare

procid1

New processor ID

partn1 Partition name of the new processor

cssid1 selected CSS ID of the new XMP processor, either appended by a # and in parenthesis or appended by a dot (.). If the CSS ID is specified for one processor only (old or new), the CSS ID for the other processor is defaulted to CSS ID 0.

procid2

Old processor ID

partn2 Partition name of the old processor

cssid2 selected CSS ID of the old XMP processor, either appended by a # and in parenthesis or appended by a dot (.)

C Indicates CSS compare report

procid1

New processor ID

cssid1 selected CSS ID of the new XMP processor, either appended by a # and in parenthesis or appended by a dot (.). If the CSS ID is specified for one processor only (old or new), the CSS ID for the other processor is defaulted to CSS ID 0.

procid2

Old processor ID

cssid2 selected CSS ID of the old XMP processor, either appended by a # and in parenthesis or appended by a dot (.)

OL Limit OS compare report by single compare reports

OS Operating system compare

ED EDT compare

GE Generic compare

GU Generic update compare

ES Esoteric compare

NI OS console compare

OD OS device compare

osid1 New operating system ID

osid2 Old operating system ID

O Indicates OS compare report

osid1 New operating system ID
osid2 Old operating system ID
SL Limit switch compare report by single compare reports

SW Switch compare
PO Switch port compare
SC Switch configuration compare
PC Port configuration compare

swid1 New switch ID
swid2 Old switch ID

S Indicates SWITCH compare report
swid1 New switch ID
swid2 Old switch ID

Print options for CSS/OS Compare Report

C Print all devices. If C is not selected, only devices are printed that are

- Defined for the CSS, but not for the OS
- Defined for the OS, but not for the CSS
- Defined for both, but of different device type

D Indicates CSS/OS compare

procid Processor ID

partn Partition name. This is a required parameter, if the processor runs in LPAR mode. For more information, see “Compare CSS / operating system views” on page 269.

osid Operating system ID

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
STEPLIB	SYS1.SCBDHENU (required for OS device compare)
HCDIODFS	New IODF
HCDIODFT	Old IODF (only for IODF compare)
HCDMLOG	HCD Message Log data set
HCDRPT	Report data set; record size 133, record format fixed block
HCDTRACE	Trace data set (if trace is activated)

Example 1:

The following example shows a job to *compare two IODFs*.

```
//BWINJOB JOB (3259,RZ-28), 'BWIN', NOTIFY=BWIN, CLASS=A,
//          MSGCLASS=Q, MSGLEVEL=(1,1), REGION=4M
//*
```

```

/** COMPARE IODFs WITH ADDED AND DELETED DATA
/** DEVICE, DEVICE ATTACHMENT AND OS DEVICE COMPARE
/** LIMITED TO LPAR PROC2.LPAR1 ON BOTH AND OS PROD17
/**
//COMPARE1 EXEC PGM=CBDMGHCP,
//          PARM='COMPARE,AB,CL,DVDA,PROC2,LPAR1,PROC2,LPAR1,
//          OL,PROD17,PROD17'
//STEPLIB DD DSN=SYS1.SCBDHENU,DISP=SHR
//HCDIODFS DD DSN=BWIN.IODF06.WORK,DISP=SHR
//HCDIODFT DD DSN=BWIN.IODF03,DISP=SHR
//HCDRPT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6650)
//HCDMLOG DD DSN=BWIN.HCD.LOG,DISP=OLD
//

```

Example 2:

The following example shows a job to *compare CSS/OS reports*.

```

//BWINJOB JOB (3259,RZ-28),'BWIN',NOTIFY=BWIN,CLASS=A,
//          MSGCLASS=Q,MSGLEVEL=(1,1),REGION=4M
/**
/** COMPARE CSS/OS CONFIGURATION BETWEEN
/** DEVICES CONNECTED TO PROC1, PART1 ON CSS SIDE
/**          DEFINED TO OS MVS1
/**
//COMPARE2 EXEC PGM=CBDMGHCP,
//          PARM='COMPARE,AB,D,PROC1,PART1,MVS1'
//HCDIODFS DD DSN=BWIN.IODF06.WORK,DISP=SHR
//HCDRPT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6650)
//HCDMLOG DD DSN=BWIN.HCD.LOG,DISP=OLD
//

```

Import an IODF

This utility function allows you to import configuration data (previously exported from another system) into an IODF. It is assumed that the mentioned configuration data has been received outside HCD, for example, using the TSO RECEIVE command, and stored in a sequential data set.

The data set containing IODF data to be imported has to be specified with DD name HCDIN.

This function is invoked by passing the following parameter string.

Import an IODF

```

|—IMPORT—, —IODF name—, —volume—, —REPLACE—

```

IODF name

Specifies the name of the target IODF (fully qualified).

volume

Specifies the volume serial number of the IODF destination. This parameter is neglected if the target IODF already exists and REPLACE is specified, or, if the data set is SMS managed.

REPLACE

Specifies that an IODF with the same name will be replaced by the received IODF. If REPLACE is not specified, the IODF is not replaced.

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
HCDIN	The data set containing IODF data to be imported
HCDMLOG	HCD Message Log data set
HCDTRACE	Trace data set (if trace is activated)

Example:

```
//BWINJOB JOB (3259,RZ-28), 'BWIN', NOTIFY=BWIN, CLASS=A,
//          MSGCLASS=Q, MSGLEVEL=(1,1), REGION=4M
// *
// * IMPORT AN IODF
// *
//IMPORT EXEC PGM=CBDMGHCP,
//          PARM='IMPORT,BWIN.IODF08,DATA04'
//HCDIN DD DSN=BWIN.EXPORTED.IODF03, DISP=SHR
//HCDMLOG DD DSN=BWIN.HCD.LOG, DISP=OLD
//
```

Export an IODF

This utility function allows you to send an IODF to another system.

You have to preallocate the IODF you want to export with DD name HCDIODFS.

If you want to send an IODF to an unattended z/OS system, you have to allocate a data set with DD name HCDCNTL. From this data set, HCD extracts information to set up the JCL to run on the unattended target system. You have to modify or adapt the JOB statement, JES routing statement(s), and JOBLIB information in this data set before you call the export utility.

This function needs to be executed in an TSO environment, for example, by invoking HCD under control of the TSO terminal monitor program IKJEFT01.

This function is invoked by passing the following parameter string.

Export an IODF

—EXPORT—, —user id—, —node id—, —IODF name—, —volume—, —
—ACTLOG—, —NOREPLACE—

userid Specifies the user ID of the target, or a nickname (nickname only if the IODF is not sent to an unattended z/OS system).

node id

Specifies the node ID of the target system, but only if the IODF is not sent to an unattended target system (otherwise it is ignored in favor of information provided by the JCL).

IODF name

Specifies the name of the target IODF. Default is the name of the source

IODF prefixed with the specified target user ID. This parameter is only applicable if the IODF is sent to an unattended z/OS system, otherwise it is ignored.

volume

Specifies the volume serial number of the DASD on which the target IODF is created if it does not exist. This parameter is only applicable if the IODF is sent to an unattended z/OS system, and the IODF data set is not managed by SMS, otherwise it is ignored.

ACTLOG

Specifies that the appropriate Activity Log file should also be sent. If this parameter is missing, or the target is an unattended MVS system, it is not sent.

NOREPLACE

For unattended exports, this keyword provides overwrite protection for an IODF at the target system with the same name as the IODF to be exported.

Batch invocation

A data set must be allocated to the following DD names when invoking the batch utility.

DD name	Description
HCDIODFS	IODF to be exported
HCDCNTL	JCL data set containing the JOB statement, the JES routing statement(s) and the JOBLIB information for sending the IODF to an unattended z/OS system.
SYSTSPRT	Print data set
SYSTSIN	SYSIN data set
HCDMLOG	HCD Message Log data set
HCDTRACE	Trace data set (if trace is activated)

You can also find the following examples in the data set SYS1.SAMPLIB.

Example 1:

The following example shows a job to export an IODF to an attended system.

```
//BWINEX1 JOB (3259,7030-83),'BWIN',CLASS=A,USER=BWIN,
//      MSGLEVEL=(1,1),NOTIFY=BWIN,MSGCLASS=Q,REGION=4M
/*
//EXPORT1 EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//HCDIODFS DD DSN=BWIN.IODF52.WORK,DISP=SHR
//HCDMLOG DD DSN=BWIN.HCD.LOG,DISP=OLD
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
          CALL 'SYS1.LINKLIB(CBDMGHCP)',      +
          'EXPORT,BMGN,BOETST1'
/*
//
```

Example 2:

The following example shows a job to export an IODF to an unattended z/OS system.

```

//BWINEX2 JOB (3259,7030-83), 'BWIN', CLASS=A, USER=BWIN,
//      MSGLEVEL=(1,1), NOTIFY=BWIN, MSGCLASS=Q, REGION=4M
//*****
/* MODIFY AND ADAPT DATA SET ALLOCATED WITH DDNAME HCDCNTL
/* BEFORE YOU SUBMIT THIS JOB.
/* USE HCDCNTL2 FOR A JES2 SYSTEM
/* USE HCDCNTL3 FOR A JES3 SYSTEM
//*****
//EXPORT1 EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//HCDIODFS DD DSN=BWIN.IODF52.WORK, DISP=SHR
//HCDMLOG DD DSN=BWIN.HCD.LOG, DISP=OLD
//HCDCNTL DD DSN=SYS1.SAMPLIB(HCDCNTL2), DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
        CALL 'SYS1.LINKLIB(CBDMGHCP)',      +
        'EXPORT,BMGN, ,BMGN.IODF11.WORK,DATA05'
/*
//

```

Example 2.1:

The following example shows the JCL statements that may be specified in a data set allocated with DD name HCDCNTL for a JES3 system.

```

//BWINEX1 JOB (3259,7030-83), 'BWIN', CLASS=A, NOTIFY=BWIN,
//      MSGCLASS=Q, MSGLEVEL=(1,1), REGION=4M
//*
/* JCL STATEMENTS SPECIFIED WITH DDNAME HCDCNTL
/*
/*ROUTE XEQ BOETST1
//BBMGNIM NJB (3259,7030-83), 'BMGN', CLASS=A,
//      MSGCLASS=Q, MSGLEVEL=(1,1), REGION=4M,
//      USER=BMGN, PASSWORD=password
//OUT1 OUTPUT JESDS=ALL, DEFAULT=YES, DEST=BOEVS01.BWIN
//

```

Example 2.2:

The following example shows the JCL statements that may be specified in a data set allocated with DD name HCDCNTL for a JES2 system.

```

//BMGNIM JOB (3259,7030-83), 'BMGN', CLASS=A,
//      MSGCLASS=Q, MSGLEVEL=(1,1), REGION=4M,
//      USER=BMGN, PASSWORD=password
/*ROUTE XEQ BOETST1
//

```

Note: You can replace the /*ROUTE statement by the /*XMIT statement.

```

/*XMIT XEQ BOETST1 DLM=xx

```

When you use the DLM parameter with the /*XMIT statement, you specify a two-character delimiter to terminate the data being transmitted. (For the end of the records to be transmitted, the default is /* in the input stream.)

Batch IODF copy example

You may want to use the batch initialize and copy IODF functions when performing system maintenance. For example, these functions can be used when copying all the data sets from one volume (that contains IODFs) to another volume.

The following sample jobs show how to copy multiple IODFs to an alternate volume. In the example it is assumed that the following IODFs exist: SYS1.IODF00 and SYS1.IODF03. The first sample job allocates corresponding data sets on volume DATA02. The second sample job initializes these data sets into IODFs, and copies the source IODFs into the newly created IODFs. Jobs similar to these are contained in member CBDSALIO and CBDSCPIO in SYS1.SAMPLIB.

```

/* JOB TO DEFINE IODF
//DEFIODF JOB REGION=4M,...
/*
/* DEFINE NEW IODF DATASETS SYS2.IODF00, SYS2.IODF03
/*
//ALLOC EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *

        DEFINE CLUSTER (NAME (SYS2.IODF00.CLUSTER) -
                        LINEAR -
                        RECORDS (1024) -
                        VOLUMES(DATA02) -
                        ) -
        DATA (NAME (SYS2.IODF00))

        DEFINE CLUSTER (NAME (SYS2.IODF03.CLUSTER) -
                        LINEAR -
                        RECORDS (1024) -
                        VOLUMES(DATA02) -
                        ) -
        DATA (NAME (SYS2.IODF03))

/* JOB TO COPY IODF
//INITIODF JOB REGION=4M,...
/*
/* INITIALIZE AND COPY SYS1.IODF00 to SYS2.IODF00
/*
//INIT1 EXEC PGM=CBDMGHCP,PARM='INITIODF SIZE=1024,ACTLOG=NO'
//HDCNTL DD *
This IODF is a copy of SYS1.IODF00
/*
//HCDIODFT DD DSN=SYS2.IODF00,DISP=OLD
//HCDMLOG DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6650)
/*
//COPY1 EXEC PGM=CBDMGHCP,PARM='COPYIODF'
//HCDIODFS DD DSN=SYS1.IODF00,DISP=SHR
//HCDIODFT DD DSN=SYS2.IODF00,DISP=OLD
//HCDMLOG DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6650)
/*
/* INITIALIZE AND COPY SYS1.IODF03 to SYS2.IODF03
/*
//INIT2 EXEC PGM=CBDMGHCP,PARM='INITIODF SIZE=1024,ACTLOG=NO'
//HDCNTL DD *
This IODF is a copy of SYS1.IODF03
/*
//HCDIODFT DD DSN=SYS2.IODF03,DISP=OLD
//HCDMLOG DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6650)
/*
//COPY2 EXEC PGM=CBDMGHCP,PARM='COPYIODF'
//HCDIODFS DD DSN=SYS1.IODF03,DISP=SHR
//HCDIODFT DD DSN=SYS2.IODF03,DISP=OLD
//HCDMLOG DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6650)
//

```

Note:

1. This batch job issues job message IEC161I, which can be ignored.

2. The VSAM DIV file consists of a data and a cluster file. According to the naming convention, the name of the data file is the IODF name (in this example SYS1.IODF01), and .CLUSTER is appended to the data file for a cluster file. Append .CLUSTER to the IODF name in the DEFINE CLUSTER statement (in this example SYS1.IODF01.CLUSTER).

When designating the number of records to be allocated in an IODF (specified on the DEFINE CLUSTER statement and as a PARM value on the INITIODF job step), it is important that the target IODF be allocated at least as big as the source IODF. While using the HCD dialog, you can use the SHOWIODF command from the command line, or the View action bar choice to display the number of records allocated in the source IODF.

Note: Remember to also copy the associated LOADxx members. For information on the LOADxx members, see *z/OS MVS Initialization and Tuning Reference* and *z/OS MVS Initialization and Tuning Guide*.

List of standard DD names

Standard DD names are used in the job control statements that define the data sets used by HCD. These names are shown in Table 24. If you want to change these names, you must create a list of alternate DD names, using the standard format for such a list.

Table 24. Standard DD names Used by HCD

	DD name	HCD Task	Data Set Contents
1	not used		
2	not used		
3	not used		
4	HCDLIB	Migration	Assembler macro library
5	HCDIN	Migration	Migration input
6	HCDPRINT	Migration	Migration log (messages)
7	not used		
8	HCDUT1	Migration	Assembler work file
9	HCDUT2	Migration Activity Log	Modified IOCP, MVSCP, and HCPRIO input to assembler Target activity log during copy
10	HCDUT3	Migration	Assembler output (object) data
11	not used		
12	HCDTERM	Migration	Assembler and loader messages
13	not used		
14	not used		
15	not used		
16	not used		
17	HCDRPT	Query/print	HCD reports
18	HCDALOG	All	Activity log
19	HCDJES3	All	JES3 initialization stream checker data
20	HCDASMP	Migration	Assembler output listing

Table 24. Standard DD names Used by HCD (continued)

	DD name	HCD Task	Data Set Contents
21	HCDDECK	Activation	IOCP and HCPRIO input data set (output)
22	HCDIODFP	All	First IODF
23	HCDIODEFS	Maintain IODF	Source IODF (for COPY, for example)
24	HCDIODFT	Maintain IODF	Target IODF
25	HCDPROF	Tailor HCD defaults	HCD profile definitions
26	HCDMLOG	All	Message log
27	HCDTRACE	All	Trace data set (if trace is activated)
28	HCDCNTL	Activation	Control file for Build IOCDs
		Maintain IODF	Control file for Build IOCP input data set JCL data set for Import/Export IODF IODF description List of configurations for Build I/O configuration data
		HMC-wide activation	TCP/IP connection table

List of alternate DD names

If used, this optional list, must start on a halfword boundary that is not also a fullword boundary.

- The first two bytes must contain a binary count of the number of bytes in the rest of the list.
- The rest of the list specifies alternate DD names that you wish to use in place of the standard DD names.
 - DD names in the alternate list must appear in the same sequence as they appear in the standard list.
 - Each name must be eight characters long. If a name contains fewer than eight characters, pad it with blanks. If you omit an alternate DD name, set that entry in the alternate DD names list to binary 0.
 - Entries in the alternate DD names list that correspond to empty entries in the standard DD names list must be set to binary 0.

————— End of Programming Interface information —————

————— End Programming Interface Information —————

Chapter 14. Security and other considerations

This information unit discusses the following topics:

- Security-related considerations
- Catalog-related considerations
- Considerations concerning data sets cataloged with an esoteric device group name
- SMS-related considerations
- ISPF and TSO/E aspects that you need to consider
- z/VM-related considerations

Security-related considerations

An appropriate resource-level security facility, such as Resource Access Control Facility (RACF*) 1.9 or an equivalent security product, is required to control access to the data sets used by HCD. You perform the access control in two steps:

1. Define the necessary RACF profiles
2. Give users access authority

Note: If no security product is installed, you cannot perform the *activate* function from HCD.

For additional RACF definitions necessary to run the HCD LDAP Backend against IODF data, see Chapter 15, “How to provide LDAP support for HCD,” on page 369.

Defining RACF profiles

You define three types of profiles:

1. Data Set Profiles

Define data set profiles for all data sets used by HCD.

2. OPERCMDS Class Profile:

Define the profile MVS.ACTIVATE to invoke the dynamic reconfiguration function under HCD, to use the MVS operator command ACTIVATE from an MVS console, or to use the HCD I/O Autoconfiguration functionality. For a description of the command syntax, see *z/OS MVS System Commands*. For a description of I/O Autoconfiguration, see Chapter 8, “How to work with I/O Autoconfiguration,” on page 187.

If you issue the ACTIVATE command, the I/O supervisor calls jobname IEASYSAS stepname IOSAS to assist in the activate procedure. IOSAS requires read access to the IODF data sets. Because the default entry for IOSAS in the Program Properties Table (PPT) is PASS, RACF checking occurs. ICH408I is the result of an ACTIVATE IODF=XX command. To ensure the successful completion of the activate process, you have to choose one of the following alternatives:

- Place the IOSAS task into the RACF started task table (ICHRIN03) and indicate that the user is authorized.
- Define the IODF data sets to RACF with UACC=READ.

- Add IOSAS as an entry in the Started Procedures Table with a valid user ID. This user ID must have read access to the SYS1.NUCLEUS and the IODF data sets.

The ACTIVATE command needs UPDATE access, regardless whether the TEST option is specified or not.

You also have to define the profile MVS.DISPLAY.IOS with read access if you wish to work from the sysplex member list to view the active configuration status or to process a CONFIGxx member.

3. FACILITY Class Profiles:

Define the following profiles:

- CBD.CPC.IPLPARM to query and update the IPLADDR and IPLPARM attribute values of the last IPL, and to be used for next IPL.
- CBD.CPC.IOCDS to query and update IOCDS control information.

To enable users to work with CPC images (see “Activate a configuration HMC-wide” on page 239), you need to define the following profile:

- HWI.TARGET.network.cpcname.* - using the BCPii community name specified with APPLDATA('community_name'). This community name must be defined on the support element of each CPC that is queried.

See also “Access to HWI.* profiles” on page 362.

Giving users access authority

The access authority you can give to a user depends on the profile.

Access to data set profiles

You can give READ, UPDATE, or ALTER access to IODFs in general or to a specific IODF.

Access to profile MVS.ACTIVATE

You must give UPDATE access to allow the user to activate a configuration change or to use the I/O Autoconfiguration function.

You can give READ access if you want to restrict the *activate* function to the test option.

Access to profile CBD.CPC.IPLPARM

NONE

Indicated that the user is not allowed to query or change the IPLADDR and IPLPARM attribute values. This is also the case if profile CBD.CPC.IOCDS is not defined or RACF is not installed,

READ Allows the user to query the IPLADDR and IPLPARM attribute values; however changing the IPLADDR and IPLPARM attribute values is not allowed.

UPDATE

Allows the user to update the IPLADDR and IPLPARM attribute values.

Table 25 on page 361 shows the relationship between HCD IPL attribute management functions and the CBD.CPC.IPLPARM access authority. Option 2.11 refers to option 2 on the *Primary Task Selection* panel and option 11 on the resulting panel.

Table 25. CBD.CPC.IPLPARM access authority and HCD IPL attribute management functions

Option	HCD IPL Attribute Management Functions	RACF Authority
2.11	List z Systems cluster	READ (or READ authority in CBD.CPC.IOCDS)
2.11	View IPL attributes	READ
2.11	Update NEXT IPL attributes	UPDATE

Access to profile CBD.CPC.IOCDS

If profile CBD.CPC.IOCDS is not defined or RACF is not installed, the local IOCDS functions (that is for processors with no SNA address specified) work as before, that is, the operator will be requested to approve the write-IOCDS request.

The new remote IOCDS functions (that is for processors with an SNA address specified) require RACF authorization.

NONE

The user is not allowed to query or change IOCDS control information, or to write an IOCDS (neither by HCD nor IOCP).

READ Allows the user to query IOCDS control information. Changing IOCDS control information or writing an IOCDS is not allowed (neither by HCD nor IOCP).

UPDATE

Allows the user to write IOCDSs (by HCD or IOCP), or to change and view IOCDS control information. If profile CBD.CPC.IOCDS is defined, then the operator will not be requested to approve the writing of an IOCDS. (That is, only users with update access to profile CBD.CPC.IOCDS are allowed to write an IOCDS.)

Table 26 shows the relationship between IOCDS management functions and the CBD.CPC.IOCDS access authority. The first column in the table refers to the options you have to select to get to the HCD functions, that is, you start with option 2 on the primary selection panel and select options 2, 6, or 11 on the resulting panel.

Table 26. CBD.CPC.IOCDS access authority and HCD IOCDS management functions

Option	HCD IOCDS Management Functions	RACF Authority
2.11	List z Systems cluster	READ (or READ authority in CBD.CPC.IPLPARM)
2.11	View IOCDS control information	READ
2.11	Update IOCDS control information (switch IOCDS, enable or disable write protection)	UPDATE
2.2 or 2.6	Build IOCDS (SNA address not defined for processor <i>or</i> batch IOCP job runs on SP 4.3 system)	UPDATE ¹ or Profile not defined to RACF ²
2.2 or 2.6 or 2.11	Build IOCDS (SNA address defined for processor <i>and</i> batch IOCP job runs on SP 5.1 system)	UPDATE ¹

Table 26. CBD.CPC.IOCDS access authority and HCD IOCDS management functions (continued)

Option	HCD IOCDS Management Functions	RACF Authority
--	Direct invocation of IOCP	UPDATE ¹ or Profile not defined to RACF ²

Note:

¹ The build IOCDS function does not require authorization by the system operator, that is, no WTOR message is written.

² A WTOR message will be issued to the operator to authorize the build IOCDS function.

For more information on security considerations for IOCDS management, refer to the *IOCP User's Guide*.

Access to HWI.* profiles

To enable users to work with CPC images as described in “Activate a configuration HMC-wide” on page 239, define access rights as follows:

- As the profile HWI.APPLNAME.HWISERV in the FACILITY class of the security product controls which applications can use BCPii services, the security administrator must give at least READ access to this resource.
- READ access is required for users of any specific CPC resource HWI.TARGET.network.cpcname.* in the FACILITY class of the security product.

For more information, refer to *z/OS MVS Programming: Callable Services for High-Level Languages*.

How to set up PassTickets for working with CPC images on z/OS

If you do not provide a password in the connection table for the remote systems, HCD uses a PassTickets for verifying the authorization for the user ID on this remote system. To allow this, your z/OS security product must support creating PassTicketss(R_GenSec) and their evaluation through the SAF interfaces. If you are using a security product other than z/OS IBM Security Server (RACF®), check with your vendor. How to configure PassTickets depends on your external security manager. The following samples show how to define PassTicketswith RACF.

To configure PassTickets support for HCD to work with CPC images, you must provide the corresponding RACF definitions on both the managing system and on the affected target systems.

If you do not provide a password in the connection table for the remote systems, HCD uses a PassTickets for verifying the authorization for the user ID on this remote system. To configure PassTickets support for HCD to work with CPC images, you must provide the corresponding RACF definitions on both the managing system and on the affected target systems.

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

A PassTickets is validated against a RACF profile name. The RACF profile name for the HCD dispatcher is CBDSERVE. You need to perform the following steps on both the managing and the target system, except of *Step 3*.

Step 1:

Before creating the necessary application profile, the RACF class PTKTDATA must be activated, if not already done:

Security Server (RACF) Example

```
SETROPTS CLASSACT(PTKTDATA)
SETROPTS RACLIST(PTKTDATA)
```

Step 2:

Then define a profile for the HCD dispatcher (CBDSERVE) with an associated encryption key. The key must be the same on both the system on which the PassTickets is to be generated (the HCD client system) and on the system on which the PassTickets is to be verified (the remote system).

Security Server (RACF) Example

```
RDEFINE PTKTDATA CBDSERVE SSIGNON([KEYENCRYPTED|KEYMASKED](<key>))
```

where <key> is a user-supplied 16-digit value used to generate the PassTickets. You can specify a value of your choice. Valid characters are 0 - 9 and A - F.

Example:

```
RDEFINE PTKTDATA CBDSERVE SSIGNON(KEYMASKED(0123456789ABCDEF))
```

Step 3: (on the managing system only)

The user calling the HCD dispatcher must have RACF permissions in order to generate PassTickets. Define a profile in the PTKTDATA class controlling access to the PassTickets services and set the universal access authority to NONE:

Security Server (RACF) Example

```
RDEFINE PTKTDATA IRRPTAUTH.CBDSEVE.* UACC(NONE) <-- all user IDs
RDEFINE PTKTDATA IRRPTAUTH.CBDSEVE.DOCU UACC(NONE) <-- a specific user ID
```

To generate PassTickets, all intended user IDs connecting to CBDSERVE need update permission to the newly created profile:

Security Server (RACF) Example

```
PERMIT IRRPTAUTH.CBDSEVE.* CLASS(PTKTDATA) ID(<user>) ACCESS(UPDATE)
```

where <user> is the user ID connecting to the HCD dispatcher CBDSERVE.

Step 4:

Finally you must activate the changes:

Security Server (RACF) Example

```
SETROPTS RACLIST(PTKTDATA) REFRESH
```

How to set up PassTickets for working with CPC images on z/VM

PassTickets are supported as an authentication method with z/VM 5.4 or later. The setup for using PassTickets on a remote z/VM system requires the same RACF configuration steps as for a z/OS remote system. Since the HCD dispatcher for z/VM uses the RACROUTE macro, the configuration steps described in *z/VM Security Server RACROUTE Macro Reference* are also required.

Providing additional security for devices

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

When a user attempts to allocate a device, the system uses SAF (the system authorization facility) to issue an authorization check. If RACF is installed, it checks a profile in the DEVICES class to determine whether the user can access the device. If the user does not have authority to access the device, the allocation fails. (Note that the system does not retry an allocation request that fails because the user is not authorized to access the device.)

Work with your RACF security administrator to set up profiles in the DEVICES class:

1. Determine your exact security requirements. Consider questions such as these:
 - Are there some devices that only a few users can use?
 - Are there some devices that all users can use?
 - Do some devices share the same security requirements?
2. Work with your RACF security administrator to assign profile names for the devices to be protected. Assign a discrete profile name to each device that has a unique security requirement. Assign a generic profile name to each device group that shares security requirements. For devices, RACF profile names include the following information:

sysid This is the system identifier, which is defined on the SYSNAME keyword in the IEASYSxx member of SYS1.PARMLIB.

Note: The system identifier is necessary only if different devices with the same device class, unit name, and device address can be attached to multiple systems and they have different security requirements. In most cases, you should specify an asterisk (*) for this qualifier.

device-class

This can be one of the following UCB device classes:

TP Teleprocessing or communications devices

UR Unit record devices

GRAPHIC

Graphic devices. These device classes are consistent with the class names used on the DISPLAY U operator command.

unit-name

This is a generic name (such as 3800) that identifies the device or devices.

Catalog considerations

One IODF can contain configuration data mirroring multiple processor or logical partition system images, but being a VSAM data set, it can be cataloged in only one catalog. Therefore, if you wish to share an IODF data set among multiple systems and each system is using a separate master catalog, you must define (in the master catalog of each system) an alias that relates to the user catalog on the DASD that is shared among the systems. Define aliases and the user catalog *before* using HCD to define IODF data sets. Figure 148 shows the recommended IODF catalog structure for IODFs.

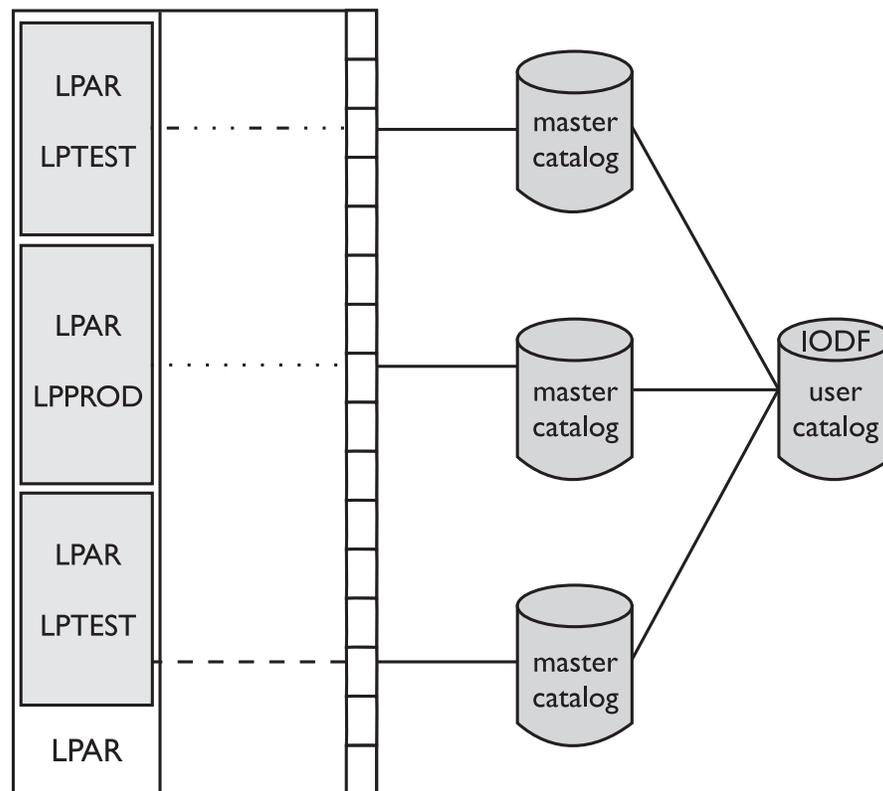


Figure 148. Recommended IODF catalog structure. (When several OSs share one IODF)

Note: It is useful to catalog the IODF in a user catalog which resides on the same volume as the IODF. That way if the volume fails and must be restored, the catalog/IODF connection is always preserved across the restore. The catalog is used to reference the IODF during HCD definition activities and during dynamic I/O reconfiguration, *not* during IPL.

Data sets cataloged with an esoteric device group name

When using HCD, data sets that were previously cataloged with an esoteric device group name (for example, SYSDA) by use of the DEFINE NONVSAM or IMPORT CONNECT command of the Integrated Catalog Facility, can cause unpredictable results if such a data set is accessed through the catalog. The reason is that the catalog entry contains the EDT-index pointing to the esoteric. The order of the esoteric in the EDT is no longer determined by the order in which the esoterics are defined, because HCD arranges the esoterics alphabetically.

To avoid this problem, you can do one of the following:

- Specify a token for the esoterics.

The esoteric token is used by allocation to find the appropriate esoteric for a data set that has been cataloged using the esoteric. You no longer have to maintain a chronological order and may delete and add esoterics without getting access problems for data sets that are cataloged using esoterics. Tokens for system built esoterics (for example, SYSALLDA) are generated by allocation and always have the same value (for SYSALLDA 9999 decimal, for example). You cannot control the token for system built esoterics. To circumvent the problem, define a new user esoteric with a token that corresponds to the EDT index in the catalog entry and that contains the same device list as SYSALLDA.

To get to the EDT index:

- Use the LISTCAT command, or,
 - If you have your MVSCP deck, count the UNITNAME statements for esoterics up to the statement that defines the esoteric name to get to the number for the token.
- Re-catalog the data sets with a generic device type name (for example, 3380), before using HCD to migrate IOCP/MVSCP data.

To determine if you have any data sets that have been cataloged with an esoteric, use the scan utility that is provided in the SYS1.SAMPLIB member IEFESOJL. This utility scans a catalog and lists the data sets that were cataloged with esoteric device group names. The prologue of this SAMPLIB member contains information on the modifications you have to make to the JCL to run the job in your installation.

SMS-related considerations

In a system managed by the storage management subsystem (SMS) you need to choose one of the following alternatives:

- The IODF data set is not managed by SMS. You can then specify the IODF volume serial number when creating an IODF.
- The IODF data set is managed by SMS. The automatic class selection (ACS) routines must be set up to automatically place the IODFs on the IODF volume. In this case SMS ignores the specified volume serial number except to pass it as a symbol to the ACS routines. The ACS routines, especially the storage group ACS routine, can use the volume serial number and the unit name to decide the SMS classes and the storage group.

Note: These considerations are important only for a production IODF that is used for IPL.

You also have to consider that HCD dynamically allocates some data sets (with fixed naming conventions). These data sets are:

- The data set used for the activity log. For more information see “Activity logging and change logging” on page 50. If an ACTLOG data set does not yet exist, HCD dynamically allocates one, using ESOTERIC system defaults (ALLOCxx of SYS1.PARMLIB, respectively the UADS entry). You have to make sure that the entries in your ACS routines do not conflict with the SMS provided defaults. For example, if your ACTLOG data set name is *not* managed by SMS, whereas your default ESOTERIC defines an SMS managed volume, an allocation error might result. If you want to use a specific volume, specify a volume serial number to allocate a new activity log in the HCD profile (see “Defining an HCD profile” on page 19).

- The data sets used by HCD for the migration of IOCP/MVSCP/HCPRIO data (HCDPRINT, HCDASMP, HCDUT1, HCDUT2, HCDUT3). For detailed information refer to “Insufficient data set sizes” on page 325.
- The data sets used when building an HCPRIO or IOCP input data set (both named HCDDECK), and the data set used when creating JES3 Initialization Stream Checker data (named HCDJES3).

ISPF-related considerations

The usual ISPF facilities are available for the HCD dialog. For example, you can:

- Suppress the display of function key assignments.
- Display panel identifiers.
- Change the position of the command line.

HCD supports the ISPF split-screen facility with the F2=Split and F9=Swap keys. So, if necessary, you can perform other ISPF operations during an HCD session. HCD cannot be used in two parallel ISPF sessions.

Compared to traditional ISPF applications, HCD enables system programmers to control a great number of hardware configuration objects by their related actions.

z/VM-related considerations

HCD allows the definition of VM operating systems and their devices including their VM-specific parameters. This is triggered by the “operating system type - VM” when defining an operating system. Figure 149 shows the panel where you can enter the operating system type.

```

Add Operating System Configuration

Specify or revise the following values.

OS configuration ID . . . . . OPSYS02_
Operating system type . . . . . VM      +

Description . . . . . z/VM operating system

OS config ID for D/R site . . _____ (generated for GDPS)

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

Figure 149. Define a VM operating system

When you attach a device to a VM operating system, the Define Device Parameters / Features panel displays the operating system-specific parameters. See Figure 150 on page 368 for an example of attaching a device to a VM operating system.

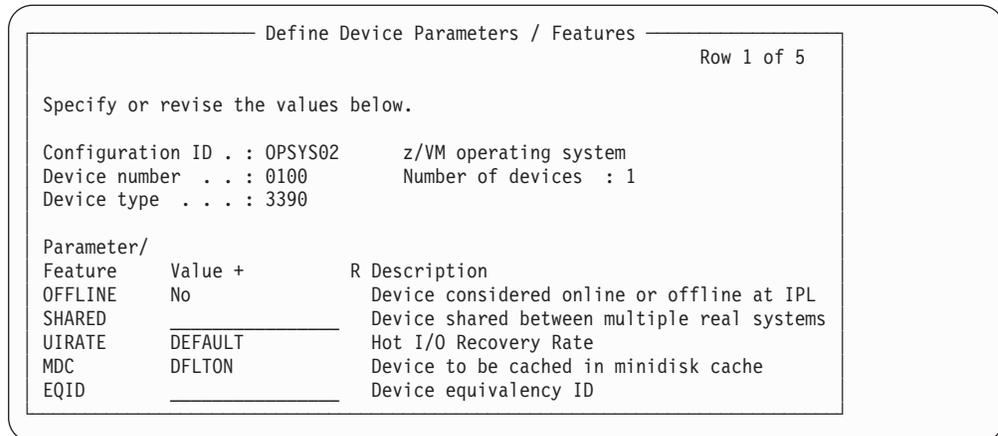


Figure 150. Define Device Parameters/Features for VM Device

The VM specific functions of HCD consist of:

- Defining an operating system of type 'VM'.
- Defining devices to a VM operating system.
- Defining VM consoles.
- Migrating an HCPRIO input data set to an IODF. For more details refer to Chapter 12, "How to migrate existing input data sets," on page 279.
- Creating an HCPRIO input data set from a production IODF. See "Build I/O configuration data" on page 226 for a description how to create an HCPRIO input data set based on the definitions in the IODF.
- Issuing VM device reports (including VM console report).

In a mixed environment, running a z/OS system in one partition and z/VM in another partition, any change of the VM definitions (for example, add a device) can be done without a POR for the processor. The *Dynamic I/O Reconfiguration* function for the hardware can be used to add this device to the Channel Subsystem (CSS). In a second step, a device can be set online dynamically. You do not have to IPL the z/VM system image. For more details refer to *z/VM CP Planning and Administration*.

If you are running z/OS and z/VM on separate processors, you can configure both systems with HCD in the same IODF. You can export the IODF to the z/VM system and activate it there using z/VM HCD. See *z/VM: I/O Configuration* for more details.

Chapter 15. How to provide LDAP support for HCD

Overview

This information unit describes:

- the LDAP structure
- the structure of the HCD LDAP backend plug-in
- how to plug the HCD LDAP backend into the IBM Tivoli Directory Server for z/OS
- how to customize the HCD LDAP backend
- how to access the IODF information
- the HCD LDAP backend operational behavior
- how to handle the LDAP requests as transactions

Before reading this section we strongly recommend that you have copies of the following documents available:

- *z/OS IBM Tivoli Directory Server Administration and Use for z/OS*
- *z/OS IBM Tivoli Directory Server Client Programming for z/OS*
- *z/OS UNIX System Services Planning*

Together with the IBM Tivoli Directory Server for z/OS and the RACF backend SDBM, the HCD LDAP backend can be used to access and update IODF data via the standardized Lightweight Directory Access Protocol (LDAP) based on TCP/IP.

The HCD LDAP backend is optional. The HCD functionality is not limited if the HCD LDAP backend is not used. If you do not want to use the HCD LDAP backend, then you do not need to read this information and you do not need to take any further actions.

All operations on IODFs are performed on behalf of user IDs which have to be explicitly permitted for the HCD LDAP backend. This does not affect your system security because the HCD LDAP backend only supports LDAP clients who are bound to the SDBM backend using a user ID and the appropriate password.

The HCD LDAP backend supports a subset of LDAP search requests and a subset of LDAP add, delete, and modify requests.

The HCD LDAP backend is able to perform sequences of update requests as transactions. The LDAP client has to support LDAP V3 controls in order to use this transaction feature.

Updates to an IODF are performed via HCD. Thus, it is ensured that the HCD validation rules are applied.

Only existing IODFs can be used with the HCD LDAP backend. The HCD LDAP backend cannot be used to create or delete IODFs. It cannot be used, for example, to build a production IODF or perform dynamic activation.

Introduction to LDAP

The following is a brief introduction to the LDAP structure and function. For more information refer to *z/OS IBM Tivoli Directory Server Administration and Use for z/OS*.

LDAP is an Internet Protocol Standard based on the TCP/IP protocol. LDAP stands for Lightweight Directory Access Protocol.

LDAP is a protocol which makes directory information accessible. A directory can be considered to be a type of yellow pages. New entries can be added, existing entries can be altered or deleted, and it is possible to search for all matching entries using wildcards.

The LDAP directory is represented hierarchically in a so-called *Directory Information Tree* (DIT). The nodes of this tree are called *entries*. Every entry is an instance of an *object class*. An object class is a name which is associated with a collection of *attributes*.

Every attribute is either mandatory (required) or optional and either single valued or multi valued. Required attributes must have one or more values, optional attributes can have zero or more values. One required single valued attribute of every object class is `objectclass`: This attribute specifies the object class of which the entry is an instance.

Each entry has a *relative distinguished name* (RDN) which is specified when the entry is created. The RDN must identify the entry uniquely among its siblings (not necessarily in the whole tree); it consists of one or more attribute-value pairs. The *distinguished name* (DN) of an entry is the sequence of the RDNs starting from the entry itself and ending with the RDN of the root entry. The DN identifies a node uniquely within the whole DIT.

An example for a DN of a partition within a DIT for HCD IODFs would be the following:

```
hcdPartitionId=PART00,hcdProcessorConfigId=PROC00,  
hcdIodfId=SYS1.IODF00,cn=HCD
```

This is illustrated in the following figure.

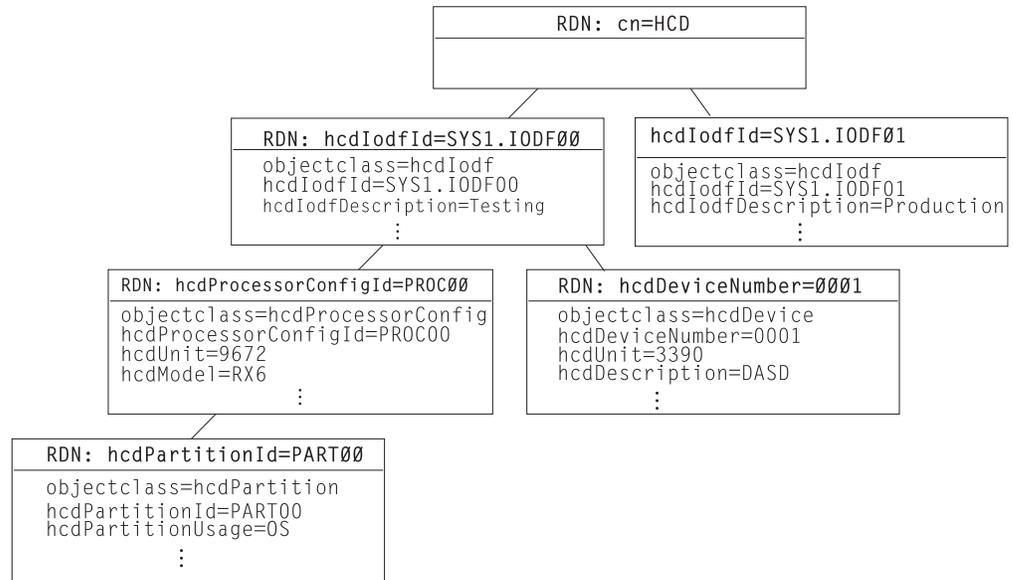


Figure 151. Sample DIT portion of the HCD LDAP backend

IBM Tivoli Directory Server for z/OS owns a DIT which can be accessed by a client via the LDAP protocol. In the case of the IBM Tivoli Directory Server for z/OS, the task of storing the DIT is delegated to several so-called *backends* or *plug-ins*. Each *plug-in* holds a certain subtree portion and is responsible for carrying out the client's requests on this portion. The DN of the root entry of a plug-in is called a *suffix*. When the IBM Tivoli Directory Server for z/OS receives a request concerning an entry it extracts the suffix from the DN of that entry in order to determine which plug-in is responsible for the request.

HCD supports the IBM Tivoli Directory Server for z/OS by providing the HCD LDAP backend plug-in which makes IODFs accessible via the LDAP protocol.

HCD LDAP backend: Structure and mode of operation

The HCD LDAP backend is plugged into the IBM Tivoli Directory Server for z/OS. It is configured using the IBM Tivoli Directory Server for z/OS configuration file (typically called `ds.conf`).

The HCD LDAP backend is similar to the RACF backend SDBM. As with SDBM, the main function of the HCD LDAP backend is to mediate between the IBM Tivoli Directory Server for z/OS and an external component, in this case HCD. HCD retains control over the IODFs; update requests are validated, processed, and the results stored by HCD in the appropriate IODF. Since it is HCD that processes the requests, updates through the IBM Tivoli Directory Server for z/OS preserve the integrity of the IODFs.

Thus, the HCD portion of the DIT must reflect the data structure of HCD exactly. For this reason, rather strict rules (as compared to the DB2 backend TDBM) have to be observed when requesting an update of IODF data through the IBM Tivoli Directory Server for z/OS.

Access control to the HCD LDAP backend is based on RACF permissions for user IDs, not (as is the usual practice) on LDAP Access Control Lists (ACLs). The HCD LDAP backend performs all services on behalf of a user ID. It accepts a service request only on condition that the associated user ID has previously been bound to

(authenticated by) SDBM. If this condition is fulfilled, the HCD LDAP backend switches to this user ID and tries to perform the request using only the RACF access rights granted to the user ID in question. In this way, access to IODFs through the LDAP interface and through the ISPF interface are both controlled by the same security mechanism. Note that this will have some consequences for the configuration of the IBM Tivoli Directory Server for z/OS.

The HCD LDAP backend uses several instances of HCD to perform operations on IODFs. Each of these instances serves exactly one request at a time on behalf of a user ID. This strategy provides an easy method of handling the validation of modified configuration data and serialization of client requests. The HCD instances are managed according to the following principles:

1. After starting up, the HCD LDAP backend opens a (configurable) minimum number of address spaces each of which contains exactly one HCD instance for handling requests. The minimum number of address spaces is controlled by the configuration file parameter **MinHcdInstances** (see “Configuration file parameters” on page 377).
2. When the HCD LDAP backend receives a legitimate request on behalf of a user ID, it assigns this request to an HCD instance. This instance is then tied to the user in question, that is, all subsequent requests from this user will be routed to this same HCD instance.
3. If the number of available instances is not sufficient, the HCD LDAP backend will open a new instance provided that a (configurable) maximum number of instances is not exceeded. The maximum number of address spaces is controlled by the configuration file parameter **MaxHcdInstances** (see “Configuration file parameters” on page 377).
4. In order that instances tied to a user can be switched to another user after having been idle for a certain time, two (configurable) timeout values can be defined:
 - The lower value specifies the time after which an instance can be switched to a user who requests a service and has not yet been tied to an HCD instance. The lower timeout value is controlled by the configuration file parameter **AllowSwitchTime** (see “Configuration file parameters” on page 377).
 - The higher value specifies the time after which the connection between an HCD instance and a user is dissolved in any case. The higher timeout value is controlled by the configuration file parameter **ForceSwitchTime** (see “Configuration file parameters” on page 377).

The lower value provides additional flexibility: As long as there is no need to switch to a new user, the current connection can be maintained until the second timeout is reached.

A special feature of the HCD LDAP backend is that it supports transactions. A transaction is a sequence of requests which is only executed as a whole. If one of the individual requests fails, the whole transaction is not carried out. This provides additional protection against inconsistency of data. Note, however, that transactions are only supported in conjunction with LDAP V3, not with LDAP V2.

Plugging the HCD LDAP backend into the IBM Tivoli Directory Server for z/OS

The HCD LDAP backend performs its services on behalf of user IDs and uses only the RACF access rights of these user IDs to determine the legitimacy of a request. This assumes that the plug-in runs under a user ID which is entitled to switch to the user ID of the respective bind request. Since the plug-in takes as its user ID

that of the IBM Tivoli Directory Server for z/OS, the HCD LDAP backend can only be plugged into the IBM Tivoli Directory Server for z/OS, if the latter runs as a started task under a user ID which is permitted to switch to another user ID. The HCD LDAP backend uses the *pthread_security_np()* service for performing this switch (thread-level security model). For more details on this switch, see the section "Preparing security for servers" of the *z/OS UNIX System Services Planning* book.

The further setup depends on which security level you choose for the IBM Tivoli Directory Server for z/OS. There are two options:

- UNIX level security
- z/OS UNIX level security

With UNIX level security, the IBM Tivoli Directory Server for z/OS must run under the superuser. The superuser has on this security level total authority over the system; in particular, he is automatically entitled to assume the identity of any other user. With z/OS UNIX level security on the other hand, the right to switch user IDs must be explicitly granted, even to the superuser.

z/OS UNIX level security is more secure than UNIX level security, and we recommend that you choose this option. However, you must be aware that this is a global decision which may have consequences for every server on your system. For this reason, the steps required for both options are described subsequently in detail. Warnings are issued whenever a step has repercussions for your system configuration.

For both options you must issue a number of RACF commands. In the example commands shown in the following descriptions, typical assumptions about the system configuration have been made. As these do not necessarily conform with your particular system configuration, you may need to modify the commands as required.

If you have already been using the IBM Tivoli Directory Server for z/OS and are adding the HCD LDAP backend, check whether your setup conforms to the requirements of the HCD LDAP backend. If it does not (for instance, you are not running the IBM Tivoli Directory Server for z/OS as a started task), then you must change your setup.

Performing the setup

It is recommended to divide the setup process into three parts. The first part consists of setting up the IBM Tivoli Directory Server for z/OS so that it is able to run with the HCD LDAP backend. The second part consists of setting up the HCD LDAP backend. The third part consists of integrating the HCD schema into the IBM Tivoli Directory Server for z/OS.

Setting up the IBM Tivoli Directory Server for z/OS

This section lists the prerequisites that the IBM Tivoli Directory Server for z/OS must comply with, so that it can run with the HCD LDAP backend.

Prerequisites for IBM Tivoli Directory Server for z/OS setup:

- Set up the IBM Tivoli Directory Server for z/OS for running as a started task.
- Establish a separate user ID that runs the IBM Tivoli Directory Server for z/OS.
- Set up the IBM Tivoli Directory Server for z/OS for running in single-server mode without replication.

- Configure the IBM Tivoli Directory Server for z/OS with the SDBM backend. The HCD LDAP backend requires the IBM Tivoli Directory Server for z/OS to run with this RACF backend. Therefore, SDBM must be included in the configuration file and all prerequisites for SDBM must be fulfilled.

Additional Prerequisites for z/OS UNIX level security:

Up to here, you have completed to set up the IBM Tivoli Directory Server for z/OS with UNIX level security. If you want to have z/OS UNIX level security, your environment must comply to the following prerequisites:

- Define the RACF FACILITY profile BPX.SERVER. Refer to *z/OS IBM Tivoli Directory Server Administration and Use for z/OS* and *z/OS UNIX System Services Planning* for details what this means for the setup of the IBM Tivoli Directory Server for z/OS.

Note: Defining a profile named BPX.SERVER in the RACF class FACILITY switches system security from UNIX level security to z/OS UNIX level security. Other applications may be affected by this switch.

- Define libraries to program control. Under z/OS UNIX level security, every program that is loaded into a server address space must be marked as controlled (see section "Defining Modules to Program Control" of the *z/OS UNIX System Services Planning* book).

Note: Changing profiles in RACF class PROGRAM can cause severe system problems if not done in a manner suitable for the system. If you are unsure ask your RACF administrator.

Setting up the HCD LDAP backend

This section describes how to set up the HCD LDAP backend. There is a sample REXX procedure CBDSLUCUS in SYS1.SAMPLIB that contains the RACF commands listed in this section for the z/OS UNIX level security.

To set up the HCD LDAP backend, perform the following steps. Steps 1 through 3 on page 375 are only required for z/OS UNIX level security. If you have chosen UNIX level security, continue with step 4 on page 376.

1. *Authorize the HCD LDAP backend to act on behalf of other user IDs.* With UNIX level security, the IBM Tivoli Directory Server for z/OS runs under a superuser (with UID 0) who is permitted to act on behalf of any user ID. Under z/OS UNIX level security, the HCD LDAP backend (which receives no password from the IBM Tivoli Directory Server for z/OS) can only perform a service for a client user ID when it has been explicitly authorized to act on behalf of that user ID (see the section "Defining Servers to Process Users without Password" of *z/OS UNIX System Services Planning*).

In order to authorize the user ID of the server to act on behalf of another user ID, you must do the following:

- a. Define a surrogate profile for the prospective client by issuing the following RACF commands (the second command updates the in-storage copy of the SURROGAT profiles):

```
RDEFINE SURROGAT BPX.SRV.userID UACC(NONE)
SETROPTS RACLIST(SURROGAT) REFRESH
```

- b. Authorize the user ID of the IBM Tivoli Directory Server for z/OS for this profile by the following commands (the second command updates the in-storage copy of the SURROGAT profiles):

```
PERMIT BPX.SRV.userID CLASS(SURROGAT) ID(LDAPSRV) ACCESS(READ)
SETROPTS RACLIST(SURROGAT) REFRESH
```

These example commands are based on the following assumptions (which may not hold for your system!):

- a. The RACF class SURROGAT has been activated.
 - b. There is no profile in that class with the name BPX.SRV.userID, where userID is the user ID of the prospective client.
 - c. The user ID of the IBM Tivoli Directory Server for z/OS is LDAPSRV.
2. *Define libraries to program control.* Under z/OS UNIX level security, every program that is loaded into a server address space must be marked as controlled (see section "Defining Modules to Program Control" of the z/OS UNIX System Services Planning book).

When using the HCD LDAP backend, the libraries containing the following load modules must be defined to program control:

Table 27. Load Module Libraries to be defined to program control

Load Modules	Typical Library
HCD LDAP backend	SYS1.LINKLIB on SYSRES
HCD	SYS1.LINKLIB on SYSRES
UIMs	SYS1.NUCLEUS on SYSRES
C++ RTL	CEE.SCEERUN on SYSRES
IBM Tivoli Directory Server for z/OS and SDBM backend	SYS1.SIEALNKE and SYS1.LPALIB on SYSRES

Note: If you use load modules from other libraries you have to define these libraries to program control as well.

To define these libraries to program control, issue the following RACF commands:

```
RDEFINE PROGRAM ** UACC(READ)
RALTER PROGRAM ** UACC(READ) ADDMEM('SYS1.LINKLIB'/'*****'/NOPADCHK)
RALTER PROGRAM ** UACC(READ) ADDMEM('SYS1.NUCLEUS'/'*****'/NOPADCHK)
RALTER PROGRAM ** UACC(READ) ADDMEM('CEE.SCEERUN'/'*****'/NOPADCHK)
RALTER PROGRAM ** UACC(READ) ADDMEM('SYS1.SIEALNKE'/'*****'/NOPADCHK)
RALTER PROGRAM ** UACC(READ) ADDMEM('SYS1.LPALIB'/'*****'/NOPADCHK)
SETROPTS WHEN(PROGRAM) REFRESH
```

The first command defines a profile named ** to the class PROGRAM. The other commands, except the last, define the libraries containing the load modules to program control. The last command refreshes the in-storage copy of the PROGRAM profiles.

The example commands are based on the following assumptions (which may not hold for your system!):

- a. The RACF class PROGRAM has been activated.
 - b. GENERIC is enabled for the RACF class PROGRAM.
 - c. There is no profile in that class with the name **.
 - d. The load modules needed reside in their typical libraries as listed Table 27.
3. When using the HCD LDAP backend, the libraries containing the following load modules must be APF authorized:

Table 28. APF authorized Load Module Libraries

Load Modules	Typical Library
HCD LDAP backend	SYS1.LINKLIB on SYSRES
C++ RTL	CEE.SCEERUN on SYSRES
IBM Tivoli Directory Server for z/OS and SDBM backend	SYS1.SIEALNKE and SYS1.LPALIB on SYSRES

The following steps apply to both security levels.

4. *Tailor the started task procedure.* This includes:
 - The HCD instances that have been started by the HCD LDAP backend have the same region size as the IBM Tivoli Directory Server for z/OS started task. So, you may need to adjust the region size of the IBM Tivoli Directory Server for z/OS started task according to the region size suitable for the HCD instances.
 - You have to ensure that the IBM Tivoli Directory Server for z/OS and the HCD LDAP backend are able to find the load modules in Table 27 on page 375 by using the z/OS search order. If the libraries containing these load modules are not searched by z/OS on your system, you must insert a STEPLIB DD, which contains the missing libraries, into the started task procedure.
5. *Tailor the IBM Tivoli Directory Server for z/OS configuration file.* You must include the definition of the HCD LDAP backend in the configuration file ds.conf. A sample of how to define the HCD LDAP backend as IBM Tivoli Directory Server for z/OS plug-in in the server configuration file is delivered with HCD and is installed in /usr/lpp/hcd/examples/ds.conf. For this purpose you must do the following (the examples are taken from the sample file):

A plug-in configuration statement must be added into the GLOBAL section of the configuration file ds.conf of the IBM Tivoli Directory Server for z/OS to define the HCD LDAP backend as a plug-in:

```
plugin clientOperation CBDMLPLG hcd_plginit "<parameters>"
```

Note: Note that the HCD LDAP backend plug-in CBDMLPLG works in 31-bit mode.

The parameters that are recognized by the HCD LDAP backend are described in "Configuration file parameters" on page 377. Here is an example of how to replace the <parameters> by acceptable keyword/value pairs (enclosed in double quotes). Note that the HCD suffix cn=HCD *must* be passed as parameter of the plug-in statement.

```
"suffix                cn=HCD
MinHcdInstances        1
MaxHcdInstances        3
AllowSwitchTime        30
ForceSwitchTime        600
TransactionRollbackTime 3600
Trace                  off
Profile                 off
TraceDsnSuffix          HCD.TRACE
ProfileDsnSuffix        HCD.PROFILE
TransformAttributeValues off"
```

Note: For the processing of the IBM Tivoli Directory Server for z/OS configuration file, the following general rules apply:

- A blank in column 1 indicates that this line is a continuation line.
- Trailing blanks are ignored.

The following syntax rules apply for specifying the parameters:

- A single parameter is defined by its keyword and its value, both separated by at least one blank.
 - Parameters are delimited by blanks; for example, you may specify each parameter in a separate line.
 - Extra imbedded blanks are ignored, but not allowed within values, for example, do not insert any blanks within `cn=HCD`.
 - Continuation lines for the plug-in statement must start in column 3 (not in column 1, as for other statements).
6. *Run the HCD LDAP backend.* To verify that your setup is working issue an LDAP request against the HCD LDAP backend. You can use the LDAP operation utilities to do this. For this purpose, enter a command according to the following template:

```
ldapsearch -h ldaphost -p ldapport -D binddn -w passwd -s base
           -b "hcdIodfId=IodfName,suffix" "objectclass=*
```

This command performs a search on the specified IODF on behalf of the user ID specified by `binddn`. `binddn` must be a DN from within the SDBM name space representing a user ID, and `passwd` the appropriate password. `IodfName` must be the name of an existing IODF data set. `suffix` would be `cn=HCD` if you have kept the default value specified in the sample configuration file `ds.conf`.

If the request returns a plausible result, the HCD LDAP backend is working correctly.

Integrating the LDAP schema for HCD

HCD is shipped with a predefined schema file representing schema definitions which the IBM Tivoli Directory Server for z/OS needs to evaluate incoming HCD requests issued via the LDAP interface. You must integrate this file into the IBM Tivoli Directory Server for z/OS after this server has been successfully installed and set up. It is recommended that this integration step is performed by the person who is responsible for the IBM Tivoli Directory Server for z/OS (usually the system administrator). The name of the HCD schema file is `schema.hcd.ldif` and is located in the `/usr/lpp/hcd/etc` directory.

Use the `ldapmodify` command to load the schema, for example:

```
ldapmodify -h ldaphost -p ldapport -D adminDN -w passwd
           -f /usr/lpp/hcd/etc/schema.hcd.ldif
```

See *z/OS IBM Tivoli Directory Server Client Programming for z/OS* for more information about `ldapmodify`.

Customizing the HCD LDAP backend

The HCD LDAP backend can be customized in the following ways:

- Parameters which are recognized by the HCD LDAP backend can be set in the IBM Tivoli Directory Server for z/OS's configuration file.
- LDAP debug levels can be set for the IBM Tivoli Directory Server for z/OS and apply to the HCD LDAP backend also.
- HCD LDAP backend can be run in English or Japanese.

Configuration file parameters

Configuration file parameters that apply to the HCD LDAP backend are set in the database section of the HCD LDAP backend in the IBM Tivoli Directory Server for

z/OS configuration file `ds.conf`. Any parameter which is not known by the IBM Tivoli Directory Server for z/OS itself is passed to the HCD LDAP backend. The following list shows the supported parameters:

MinHcdInstances *int*

Specifies the minimum number of HCD instances started during startup of the HCD LDAP backend.

Default:

1

Range:

1-10

Constraints:

Must be less than or equal to the numerical value of `MaxHcdInstances`

For example, `MinHcdInstances 1` causes the HCD LDAP backend to start exactly one initial HCD instance.

MaxHcdInstances *int*

Specifies the maximum number of HCD instances which can be used simultaneously by the HCD LDAP backend. The HCD LDAP backend will start new HCD instances dynamically if they are needed and if the maximum number has not already been reached.

Default:

3

Range:

1-10

Constraints:

Must be greater than or equal to the numerical value of `MinHcdInstances`

For example, `MaxHcdInstances 3` will allow the HCD LDAP backend to use a maximum of three HCD instances simultaneously.

AllowSwitchTime *int*

Specifies the idle time in seconds after which the user ID of an HCD instance may be changed.

Default:

30

Range:

1-120

Constraints:

Must be less than the numerical value of `ForceSwitchTime`.

For example, `AllowSwitchTime 30` will ensure that an HCD instance must be idle for at least thirty seconds before the HCD LDAP backend is allowed to switch the HCD instance to a different user ID.

ForceSwitchTime *int*

Specifies the idle time in seconds after which the user ID of an HCD instance is unconditionally reset. If a new request on behalf of this user ID arrives, this or any other free HCD instance has to be switched to that user ID.

Default:

600

Range:

60-3600

Constraints:

Must be greater than the numerical value of AllowSwitchTime.

Exception:

If the value specified for ForceSwitchTime is 0, a forced reset of the user ID will not be performed.

For example, ForceSwitchTime 600 will ensure that the user ID of an HCD instance is reset by the HCD LDAP backend after an idle time of 600 seconds (10 minutes) of the HCD instance

TransactionRollbackTime *int*

Specifies the idle time (in seconds) of an LDAP client after which a transaction, running on behalf of this LDAP client is rolled back automatically.

Default:

3600

Range:

10-3600

For example, TransactionRollbackTime 3600 will ensure that the HCD LDAP backend automatically rolls back a transaction which is still open on behalf of an LDAP client which has been idle for 3600 seconds (one hour).

Trace on | off

Determines whether the HCD instance running on behalf of the HCD LDAP backend generates an HCD trace (see "HCD trace facility" on page 488). Each HCD instance generates its own trace.

Default:

off

TraceDsnSuffix *name*

Specifies the suffix of the data set names where the traces of the HCD instances are to be written to. The effective name of the trace data set of one particular HCD instance is determined by concatenating the user ID associated with the HCD instance with the suffix.

Default:

HCD.TRACE

Constraints:

Only valid data set names with a maximum of 35 characters can be used. Note that the suffix must not be empty.

For example, assume you have set the TraceDsnSuffix to LDAP.HCD.TRACE and that you have specified Trace on. If an HCD instance now performs an LDAP request on behalf of user ID TEST and is, therefore, switched to this user ID, it will open the trace data set with name 'TEST.LDAP.HCD.TRACE' and DISP=OLD. This data set is then used for tracing the operations of the HCD instance.

Profile on | off

Determines whether the HCD instances running on behalf of the HCD

LDAP backend will use an HCD profile on startup or user ID switch (see “Defining an HCD profile” on page 19). Each HCD instance uses its own profile, depending on the user ID which the HCD instance is currently related to.

Default:

off

ProfileDsnSuffix *name*

Specifies the suffix of the profile data set name used by the HCD instances. The effective name of the profile data set of one particular HCD instance is determined by concatenating the user ID associated with the HCD instance with the suffix.

Default:

HCD.PROFILE

Constraints:

Only valid data set names with a maximum of 35 characters can be used. Note that the suffix must not be empty.

For example, assume you have set ProfileDsnSuffix to LDAP.HCD.PROFILE and you have specified Profile on, If an HCD instance now performs an LDAP request on behalf of user ID TEST and is thus switched to this user ID, it will open the profile data set with name 'TEST.LDAP.HCD.PROFILE' and DISP=SHR. The contents of the data set will be read and the HCD instance will be set up accordingly.

TransformAttributeValues on | off

Specifies whether the values of the attributes hcdIodfDescription and hcdDescription are interpreted as being in IBM-939 or not. In general, these values are interpreted as being in IBM-037. It is important to have the correct setting because the IBM Tivoli Directory Server for z/OS communicates with LDAP clients using UTF-8 representation and the HCD LDAP backend has to convert attribute values appropriately.

Default:

off

LDAP debug level

For a description of the debug levels see “LDAP problem determination” on page 484.

Message translation

The HCD LDAP backend supports English and Japanese messages. To obtain English messages with the character representation IBM-1047, choose one of the following values for the environment variable LANG:

- C
- En_US
- En_US.IBM-1047

To obtain Japanese messages with the character representation IBM-939, choose one of the following values for the environment variable LANG:

- Ja_Jp
- Ja_Jp.IBM-939

For the values of LANG supported by the IBM Tivoli Directory Server for z/OS and for more information on how to specify the value of LANG, see the *z/OS IBM Tivoli Directory Server Administration and Use for z/OS*.

Note: The settings of the environment variable LANG do not control the language of the HCD messages which HCD LDAP backend returns to LDAP clients upon their requests. In fact, all HCD messages returned to LDAP clients will always be in English.

Accessing IODF information

The HCD LDAP backend provides access to IODF information via the LDAP protocol. You can, for instance, add or delete devices and control units, or modify operation system settings. It is also possible to automate complex updates, for instance a whole I/O configuration, by communicating with the IBM Tivoli Directory Server for z/OS through an application (see *z/OS IBM Tivoli Directory Server Client Programming for z/OS*).

The IODF directory information tree

In order to make HCD IODF information accessible via the LDAP protocol, the HCD LDAP backend realizes a one-to-one mapping between the IODF data structure and the structure of the LDAP directory information tree (DIT). It is this one-to-one mapping that makes the update of IODFs through the IBM Tivoli Directory Server for z/OS possible.

The resulting HCD portion of the DIT has some special constraints as compared to an all purpose backend like TDBM:

- One important difference between the HCD portion and TDBM concerns the relation between object class and position in the (sub)tree. In TDBM, every object class can occur on every position. The HCD subtree, on the other hand, must correspond to the structure of the IODF. Accordingly, the object class of every entry managed by the HCD LDAP backend uniquely determines the object class of its parent entry.

Note, however, that this parent relation cannot be understood as in object oriented programming: Thus the object class `hcdPartition` has as its parent the object class `hcdProcessorConfig`, but `hcdPartition` is certainly not derived from `hcdProcessorConfig`.

- In addition, there are a number of HCD specific dependencies between entries that reside on different branches of the HCD subtree. In some cases, the HCD LDAP backend takes these dependencies into account by automatically adding or deleting certain entries when certain other entries are added or deleted. In other cases you must ensure yourself that your request conforms with these interrelationships. For details, see “Restrictions for search and update requests” on page 383 and Appendix F, “IODF data model,” on page 511.
- The schema of the HCD portion of the DIT is fixed. It cannot be extended or modified.

The object classes required for the IODF DIT and their interrelationships (parent relation and branch crossing interrelationships) are described in Appendix F, “IODF data model,” on page 511. This appendix also contains descriptions of all the attributes that are needed for these object classes.

Performing LDAP requests on IODFs

In order to request a service from the HCD LDAP backend you must always perform two steps:

1. *Authenticate yourself to the RACF Backend.* For authentication, you have to specify a so-called *bind* DN and a password. The authentication (called *binding* in LDAP terminology) is successful when the bind DN identifies an entry of the backend to which the suffix of the bind DN is assigned, and when the specified password is in fact associated with this entry. The HCD LDAP backend does not perform this identity check itself, but uses the SDBM backend for that purpose. Thus, you must specify a user ID in the bind DN, and the bind DN must have the following structure:

```
"racfid=user_ID,profileType=user,sysplex=Your_Sysplex,..."
```

Here *sysplex=Your Sysplex* is the first element of the suffix. Note that group is not allowed as the value of the *profileType* attribute.

If the binding to the RACF Backend fails, the HCD LDAP backend will not support the subsequent requests.

2. *Access an IODF.* If you want to access an IODF with the data set name *IodfDsn*, the DN in your LDAP request must have the following form:

```
". . . ,hcdIodfId=IodfDsn,suffix_of_HCD_LDAP_Backend"
```

Note that only IODFs that can be reached by the IBM Tivoli Directory Server for z/OS are accessible with the HCD LDAP backend.

If you have issued your request as described, the HCD LDAP backend will take the following actions:

- The user ID is extracted from the initial bind request.
- The IODF data set name is extracted from the LDAP request.
- A prepared address space is switched to the user ID, and HCD is started.
- HCD loads the specified IODF and performs the requested operations.
- The results are sent back to you.

Operational behavior

The operational behavior of the HCD LDAP backend is basically the same as that for the RACF backend with some small differences. Information about the RACF Backend can be obtained from the *z/OS IBM Tivoli Directory Server Administration and Use for z/OS*.

The following gives an overview of the functional behavior of the HCD LDAP backend:

1. The HCD LDAP backend does not participate in extended group membership searching on a client request.
2. It is possible to run several HCD LDAP backend on one IBM Tivoli Directory Server for z/OS simultaneously.
3. The root of a subtree (see Figure 151 on page 371) is denoted by a suffix in the configuration file. You can specify only one suffix per HCD LDAP backend. Suffix names must be unique if you are running multiple HCD LDAP backend.
4. The HCD LDAP backend does not support Access Control Lists (ACLs) which are normally used to protect information stored in an LDAP directory from

unauthorized access. The reason for this is that the DIT portions managed by the HCD LDAP backend are based on IODF data sets for which all access control is performed by RACF.

5. The HCD LDAP backend does not support the following LDAP request types and will answer these requests with the return code "Unwilling to Perform":
 - Bind
 - ModifyDN (also called ModifyRDN, or ModRdn)
 - Compare
 - Abandon
 - Extended Request
6. The following LDAP request types are supported by the HCD LDAP backend:
 - Add
 - Delete
 - Modify
 - Search

The following table shows how the HCD LDAP backend behaves during these LDAP operations:

Target DN	Search	Add	Delete	Modify
<i>suffix</i>	Error: Inappropriate Matching	Error: Inappropriate Matching	Error: Inappropriate Matching	Error: Inappropriate Matching
<i>hcdIodfId= IodfDsn,suffix</i>	Perform the appropriate search request. See "Search"	Error: Inappropriate Matching	Error: Inappropriate Matching	Perform the appropriate modify request. See "Modify" on page 386
<i>...,hcdIodfId= IodfDsn,suffix</i>	Perform the appropriate search request. See "Search"	Perform the appropriate add request. See "Add" on page 384	Perform the appropriate delete request. See "Delete" on page 385	Perform the appropriate modify request. See "Modify" on page 386

7. Multi-server or replication is not supported by the HCD LDAP backend

Restrictions for search and update requests

This section describes the restrictions which the HCD LDAP backend imposes on the search, add, delete and modify capabilities of LDAP. Many of these restrictions derive from the fact that the structure of HCD portion of the DIT is much more rigidly controlled than, for instance, the TDBM subtree.

Note: Within a single request, references to an attribute name must either always be with the alias name or always with the full attribute name. A mix is not accepted.

In the following subsections, *suffix* stands for the suffix of the HCD LDAP backend.

Search

Searching is restricted as follows:

- Only search bases ending with *hcdIodfId=Iodf_dataset_name,suffix* are supported. This implies that only one IODF can be searched at a time.

- The only search filters that are supported by the HCD LDAP backend are `objectclass=*` and `objectclass=name`, where *name* has to be the name of an object class that is defined for the HCD LDAP backend.
- Time or size limits are not supported.
- Controls are not supported.
- It is not possible to restrict the attributes of the matching entries that will be displayed. Every attribute that has at least one value will be shown in the search results.

Example

Following are two examples for retrieving information from an existing IODF with the command line search utility of LDAP.

The command

```
ldapsearch -D "racfid=TEST,profiletype=user,sysplex=sysplex1" -w "passwd"
-s base -b "hcdIodfId=TEST.IODF00.WORK,suffix" "objectclass=hcdIodf"
```

retrieves the top entry of object class `hcdIodf` belonging to the IODF named `TEST.IODF00.WORK` on behalf of user ID `TEST`. The result may look as follows:

```
hcdIodfId=TEST.IODF00.WORK,suffix
objectClass=hcdIodf
hcdIodfId=TEST.IODF00.WORK
hcdIodfType=W
hcdIodfDescription=Testing purposes
hcdBlocksAllocated=20
hcdBlocksUsed=2
hcdCreationDate=1999-10-04
hcdLastUpdateDate=1999-12-16
hcdLastUpdateTime=09:25:50
1 matches
```

The same result could be obtained with the search filter `"objectclass=*"`.

The command

```
ldapsearch -D "racfid=TEST,profiletype=user,sysplex=sysplex1" -w "passwd"
-s one -b "hcdIodfId=TEST.IODF00.WORK,suffix" "objectclass=hcdDevice"
```

retrieves all entries of object class `hcdDevice` belonging to the IODF named `TEST.IODF00.WORK`, again on behalf of user ID `TEST`. One of the retrieved entries may look as follows:

```
hcdDeviceNumber=000D,hcdIodfId=TEST.IODF00.WORK,suffix
objectClass=hcdDevice
hcdDeviceNumber=000D
hcdUnit=2540P
hcdModel=1
hcdDescription=Virt. Puncher
```

Note: Attribute names in the search results may be in lower case only, depending on the set up of the IBM Tivoli Directory Server for z/OS for example, `hcdiodfid` instead of `hcdIodfId`. Also, there is no specific order of the attribute/value pairs in the returned result.

Add

Adding an entry is restricted as follows:

- Entries can only be added below `hcdIodfId=...,suffix`, that is, add is not supported on DN `hcdIodfId=...,suffix` or DN `suffix`.

- Since every object class of the HCD LDAP backend except `hcdIodf` has a uniquely determined parent class, ensure that the object class of the new entry and that of the entry to which the new entry is appended are related as child and parent. Exactly one value must be specified for the `objectclass` attribute. See Appendix F, “IODF data model,” on page 511 for parent-child relationships between object classes.
- If the RDN of the entry to be added is `attribute=value`, `value` must be specified as a value of `attribute` inside the entry.
- The attributes which are contained in the entry's RDN are determined by the object class of an entry. See Appendix F, “IODF data model,” on page 511.
- There must be no entry in the DIT with the same DN as the entry to be added.
- If an add request fails because of a missing parent, the HCD LDAP backend does not update the matched DN field of the result.
- Check Appendix F, “IODF data model,” on page 511 to see which object classes can be added.
- Adding an entry may cause other entries to be created automatically using default values. See Appendix F, “IODF data model,” on page 511.
- Two controls are supported for the LDAP add request. See “Transactions” on page 387 for details.

Example:

A new entry of the object class `hcdControlUnit` of type 3990 with a control unit number of 0100 can be appended to the entry `hcdIodfId=TEST.IODF00.WORK,suffix` as follows.

First create a data set member named `TEST.LDIF(ADDCU100)` with the content

```
dn:hcdControlUnitNumber=0100,hcdIodfId=TEST.IODF00.WORK,suffix
changetype:add
objectclass:hcdControlUnit
hcdControlUnitNumber:0100
hcdUnit:3990
```

Then call the LDAP command line utility `ldapadd` with the following parameters:

```
ldapadd -D "racfid=TEST,profiletype=user,sysplex=sysplex1" -w "passwd"
-f //'TEST.LDIF(ADDCU100)'
```

The entry will be added on behalf of the user ID `TEST`.

You can then verify that the entry was created correctly by issuing

```
ldapsearch -D "racfid=TEST,profiletype=user,sysplex=sysplex1" -w "passwd"
-s base -b "hcdControlUnitNumber=0100,hcdIodfId=TEST.IODF00.WORK,suffix"
"objectclass=*"
```

The search result should look like:

```
hcdcontrolunitnumber=0100,hcdIodfId=TEST.IODF00.WORK,suffix
objectclass=hcdControlUnit
hcdcontrolunitnumber=0100
hcdunit=3990
1 matches
```

Delete

Deleting an entry is restricted as follows:

- Only entries below DN `hcdIodfId=...,suffix` can be deleted. Delete on DN `hcdIodfId=..., suffix` or DN `suffix` is not supported.

- Check Appendix F, “IODF data model,” on page 511 to see which object classes can be deleted.
- Deleting one entry may cause other entries to be deleted automatically. See Appendix F, “IODF data model,” on page 511.
- Two controls are supported for the LDAP delete request. See “Transactions” on page 387 for details.

Example:

To delete the entry added in the example shown in “Add” on page 384 you can call the LDAP command line utility `ldapdelete` with the following parameters:

```
ldapdelete -D "racfid=TEST,profiletype=user,sysplex=sysplex1" -w "passwd"
"hcdControlUnitNumber=0100,hcdIodfId=TEST.IODF00.WORK,suffix"
```

The entry will be deleted on behalf of the user ID TEST.

Modify

Modifying an entry is restricted as follows:

- Only the entry DN `hcdIodfId=...,suffix` and below can be modified. Modification of DN `suffix` is not supported.
- Check Appendix F, “IODF data model,” on page 511 to see which object classes can be modified.
- The HCD LDAP backend only supports the delete and replace subcommands of modify. The add subcommand is NOT supported.
- The value of the object class attribute cannot be deleted or replaced.
- The value(s) of the attributes which are contained in the entry's RDN cannot be deleted or replaced.
- One modify request to a single entry can contain a sequence of delete and replace subcommands. This sequence can be considered as atomic: Either the whole sequence is performed or nothing is performed.
- One attribute can only be referenced once in the whole modify request. It can only be deleted once, replaced once, and only *either* be deleted *or* replaced.
- Modify delete only supports the deletion of all values of an attribute. For this reason, you must not specify values in the modify delete request. If a value is specified, the whole modify request is rejected by the HCD LDAP backend.
- Attributes described as mandatory in an object class must not be deleted
- Modify replace replaces all existing values of the given attribute with the new values listed, creating the attribute if it did not already exist. A replace with no value will delete the entire attribute if it exists, and is ignored if the attribute did not exist.
- All values must conform with the type specified in the attribute definition.
- Modifying an entry may cause other entries to be modified automatically. See Appendix F, “IODF data model,” on page 511.
- Two controls are supported for the LDAP modify request. See “Transactions” on page 387 for details.

Example:

The entry created in “Add” on page 384 can be modified by adding the attribute `hcdDescription` as follows.

First create a data set member named TEST.LDIF(REPCU100) with the content

```
dn:hcdControlUnitNumber=0100,hcdIodfId=TEST.IODF00.WORK,suffix
changetype:modify
replace:x
hcdDescription:New description
```

Then call the LDAP command line utility `ldapmodify` with the following parameters:

```
ldapmodify -D "racfid=TEST,profiletype=user,sysplex=sysplex1" -w "passwd"
-f //'TEST.LDIF(REPCU100)'
```

If the modify request completes successfully, the entry will look like:

```
hcdControlUnitNumber=0100,hcdIodfId=TEST.IODF00.WORK,suffix
objectClass=hcdControlUnit
hcdControlUnitNumber=0100
hcdUnit=3990
hcdDescription=New description
```

This `hcdDescription` can now be deleted again with the delete subrequest of modify. To do this, first create a data set member named TEST.LDIF(DELCU100) with the content

```
dn:hcdControlUnitNumber=0100,hcdIodfId=TEST.IODF00.WORK,suffix
changetype:modify
delete:hcdDescription
-
```

Then issue the following command:

```
ldapmodify -D "racfid=TEST,profiletype=user,sysplex=sysplex1" -w "passwd"
-f //'TEST.LDIF(DELCU100)'
```

Transactions

Transactions is a concept in which individual LDAP requests are handled collectively as a single unit such that if one of the requests within the transaction should fail, then the whole transaction with all its requests is not carried out. In this way, you do not have a situation where some requests are performed and others are not. Transactions can be helpful if you want to perform complex tasks which consist of LDAP request sequences. For example, if you want to define a coupling facility (which would require adding two entries of object class `hcdChannelPath`) you can perform the necessary LDAP requests within a transaction.

In general, arbitrary sequences of at least two update requests (i.e. add, delete, modify) can be performed as a transaction. A transaction cannot consist of a single update request as a single request by definition behaves like a transaction.

Prerequisites and method of functioning

The LDAP client marks a request for the HCD LDAP backend as being part of a transaction by specifying an LDAP V3 control for the request. For this reason, transactions in the HCD LDAP backend can only be used with LDAP clients supporting LDAP V3 controls.

LDAP V3 controls represent additional information which can be included with a request to, or to a response from the IBM Tivoli Directory Server for z/OS.

In order to use transactions offered by the HCD LDAP backend, the following two LDAP V3 controls, both of which are supported by the HCD LDAP backend, are required. Both controls must always be used with a valid value which determines how the HCD LDAP backend will react to the control.

Examples of how to use these controls are shown in “How to initiate, extend and close a transaction” on page 389

1. **hcdTransactionControl:**

- **Name:** **hcdTransactionControl**
- **Description:** This control is used on the first and last request of a transaction, to indicate either the start (via value NEW) and finish (via value COMMIT or ROLLBACK) of a transaction request sequence.
- **Assigned Object Identifier:** 1.3.18.0.2.10.3
- **Target of Control:** Server
- **Control Criticality:** Critical
- **Values:** Value is exactly one char 0 terminated string in UTF-8 encoding representing exactly one of the strings (words):

NEW
COMMIT
ROLLBACK

These strings are not case sensitive. The following byte values (shown as two digit hexadecimal numbers) represent these strings:

Value	Byte sequence
NEW	= 4E 45 57 00
COMMIT	= 43 4F 4D 4D 49 54 00
ROLLBACK	= 52 4F 4C 4C 42 41 43 4B 00

2. **hcdTransactionId:**

- **Name:** **hcdTransactionId**
- **Description:** This control is used on all requests of a transaction except the first request. The value of this control indicates which particular transaction a request belongs to.

If an LDAP client initiates a new transaction with an appropriate request, the HCD LDAP backend answers the request with an LDAP response containing control `hcdTransactionId`. The LDAP client has to extract this control's value from the response and must specify the value for `hcdTransactionId` on all following requests belonging to the transaction.

- **Assigned Object Identifier:** 1.3.18.0.2.10.4
- **Target of Control:** Server
- **Control Criticality:** Critical
- **Values:** Value is exactly one char 0 terminated string in UTF-8 encoding representing a non-negative, non-zero long int value in decimal format which is the transaction ID. Only values previously received from the HCD LDAP backend are allowed - all others are rejected. Example: Assuming a transaction has an ID of 238. Then, the following byte values (shown as two digit hexadecimal numbers) represent the appropriate `hcdTransactionId` control value:

32 33 38 00.

The 10 decimal digits have the following character representation in UTF-8 (shown as two digit hexadecimal numbers):

0 = 30, 1 = 31, 2 = 32, 3 = 33, ..., 9 = 39.

In order to perform a transaction containing a sequence of LDAP requests R_1, R_2, \dots, R_n , the LDAP client has to do the following:

1. Send the LDAP requests R_1, R_2, \dots, R_n of the transaction one after another to the IBM Tivoli Directory Server for z/OS and wait for a response to each request before sending the next.
2. To initiate a new transaction (containing the requests R_1, R_2, \dots, R_n) extend the first request R_1 with control `hcdTransactionControl` and specify `NEW` as value for this control. Control `hcdTransactionId` must not be used on the first request R_1 . If the HCD LDAP backend is able to open the new transaction, it will respond by returning the control `hcdTransactionControl` and the control `hcdTransactionId` with a transaction ID as *value*. In the case of failure neither controls will be returned.
3. Send all subsequent requests R_2, \dots, R_n of the current transaction, with the control `hcdTransactionId` containing the valid transaction ID issued by the HCD LDAP backend as a response to the first request R_1 .
4. Commit (or if necessary rollback) the whole transaction using the control `hcdTransactionControl` and the *value* `COMMIT` (or `ROLLBACK`). This must be added to the last request (R_n) of the transaction. If, however, any request in between fails, you can use the `hcdTransactionControl` to initiate an immediate `ROLLBACK`. The control `hcdTransactionId` with the appropriate value must, of course, also be specified.

Note: Only update requests (i.e. add, delete and modify) can be part of a transaction.

All operations belonging to a transaction must act on the same IODF and must have been issued by the same LDAP client with the same LDAP handle bound to the same user ID.

The following section shows how to initiate a transaction, to add further LDAP requests to a transaction and to close a transaction.

How to initiate, extend and close a transaction

To use the transaction facility, you have to:

- Set up and run the IBM Tivoli Directory Server for z/OS and the HCD LDAP backend as described previously in this information unit.
- Provide an LDAP V3 client program which uses the appropriate controls of the HCD LDAP backend.

Refer to *z/OS IBM Tivoli Directory Server Administration and Use for z/OS*, *z/OS IBM Tivoli Directory Server Client Programming for z/OS*, and the IBM redbook *Understanding LDAP* for examples of LDAP client programs.

The following shows some examples of how to use the previously introduced controls on LDAP requests to take the following actions:

- Initiate a transaction
- Submit further transaction requests
- End a transaction

All the following examples written in C, are provided for the LDAP add request (here, we use the LDAP request `ldap_add_ext()` from the LDAP client API in C).

Please note that you must choose the version of the LDAP client API function which allows you to specify server controls. See the *z/OS IBM Tivoli Directory Server Client Programming for z/OS* for more information on the functions themselves, as well as on the parameters which have to be passed for particular requests.

Example 1: Initiate a new transaction

A transaction is initiated using the `hcdTransactionControl` with `NEW` as value. This control can be defined the following way:

```
static LDAPControl hcdTransactionControl_new =
    {
        "1.3.18.0.2.10.3", /* -- hcdTransactionControl -- */
        { 3, "\x4E\x45\x57\x00" }, /* -- NEW ----- */
        LDAP_OPT_ON          /* -- critical ----- */
    };
static LDAPControl *hcdTC_new[2] = { &hcdTransactionControl_new, NULL };
```

Note, as mentioned before, the value `NEW` (and also the values `COMMIT` and `ROLLBACK`, shown in example 3) have to be specified using UTF-8 encoding.

All controls to be passed to the IBM Tivoli Directory Server for z/OS are stored in an array. In this case, only one control is in the NULL-terminated array.

This array is now be passed to the function which sends the appropriate request to the IBM Tivoli Directory Server for z/OS. For example, `ldap_add_ext()` is used to request an add operation as follows:

```
rc = ldap_add_ext( ld, dn, pmods, hcdTC_new, NULL, &msgidp);
```

Here, the control `hcdTC_new` is used where the *value* `NEW` was specified in UTF-8.

If this request was successful and if a new transaction was started, the HCD LDAP backend sends back the control `hcdTransactionId` which contains the transaction ID. Such control may look similar to the following (see also Example 2):

```
static LDAPControl hcdTransactionId =
    {
        "1.3.18.0.2.10.4", /* -- hcdTransactionId ----- */
        { 1, "\x31\x00" }, /* -- TXN Id ----- */
        LDAP_OPT_ON          /* -- critical ----- */
    };
static LDAPControl *hcdTC_Id[2] = { &hcdTransactionId, NULL };
```

In the previous example, for the control `hcdTransactionId` the value indicates a transaction ID of 1.

Note, you should never generate a value for this control on your own. Instead, call the LDAP client API functions `ldap_result()` and `ldap_parse_result()` on the response of the first request to obtain the valid transaction ID.

After having issued `ldap_add_ext()`, calling `ldap_parse_result()` parses the results which were previously obtained by `ldap_result()`. Here, among other parameters, the LDAP control containing the transaction ID is provided.

The following code example demonstrates how to use both LDAP requests:

```
rc = ldap_result(ld, msgidp, 0, NULL, &LDAP_TXN_Msg);
rc = ldap_parse_result(ld, LDAP_TXN_Msg, &errcodep, &matcheddn,
                      &errmsgp, &referralsp, &servctrisp, freeit);
```

The interesting parameter as far as controls are concerned is `servctrlsp`. This pointer locates an array such as `hcdTC_new`. The appropriate values of the control have to be copied into the `hcdTransactionId` control which can be done in the following way (assuming that `servctrlsp[0]` contains the `hcdTransactionId` control and the control `servctrlsp[0]` is not freed):

```
hcdTransactionId.ldctl_value.bv_val = (servctrlsp[0])->ldctl_value.bv_val;
hcdTransactionId.ldctl_value.bv_len = (servctrlsp[0])->ldctl_value.bv_len;
```

After these values have been copied, the correct transaction ID can be provided for further LDAP requests which are part of this transaction. This is shown in "Example 2: Submit further LDAP requests of the transaction."

Example 2: Submit further LDAP requests of the transaction

After a transaction was successfully initiated, further LDAP requests can be added to the transaction. Here, the user must provide an `hcdTransactionId` control where *value* contains the correct transaction ID. As stated before, this control is provided by the HCD LDAP backend and must be used for further requests belonging to this specific transaction.

If we assume that for the transaction we just initiated, the transaction ID number is 1, then, the control which was provided by the HCD LDAP backend would be as follows:

```
static LDAPControl hcdTransactionId =
    {
        "1.3.18.0.2.10.4", /* -- hcdTransactionId ----- */
        { 1, "\x31\x00" }, /* -- TXN Id ----- */
        LDAP_OPT_ON        /* -- critical ----- */
    };
static LDAPControl *hcdTC_Id[2] = { &hcdTransactionId, NULL };
```

In general, the values 1 and "\x31\x00" have to be replaced by the correct values of the control that is provided by the call of `ldap_parse_result()` (see example 1).

After the correct values for this transaction have been copied into the previous control, further LDAP requests can be added to the transaction using this control. The call of such an LDAP request is just the same as in example 1 with the only exception that the controls have been exchanged as follows:

```
rc = ldap_add_ext( ld, dn, pmods, hcdTC_Id, NULL, &msgidp);
```

As the transaction ID does not change for a specific transaction, all further requests belonging to this transaction must use the same `hcdTransactionId` control.

Example 3: End a transaction

Here, two different cases must be considered. As stated before, a transaction can be committed for execution, or it can be aborted by the user using a rollback request.

It is recommended that you explicitly rollback a transaction if a single LDAP request returns a bad return code. The reason for this is that, transactions are intended to represent logical units of requests which belong together. In principal, both the actions - commit and rollback - are the same from an implementation point of view. The only difference is, which control is to be specified for the LDAP request that finishes the transaction. If the user wants to commit a transaction, the following control must be defined:

```

static LDAPControl hcdTransactionControl_commit =
    {
        "1.3.18.0.2.10.3", /* -- hcdTransactionControl -- */
        { 6, "\x43\x4F\x4D\x4D\x49\x54\x00" }, /* - COMMIT - */
        LDAP_OPT_ON /* -- critical ----- */
    };
static LDAPControl *hcdTC_commit[3] = { &hcdTransactionControl_commit,
                                        &hcdTransactionId, NULL };

```

If the user want to rollback a transaction, the following control must be defined:

```

static LDAPControl hcdTransaction_rollback =
    {
        "1.3.18.0.2.10.3", /* -- hcdTransactionControl -- */
        { 8, "\x52\x4F\x4C\x4C\x42\x41\x43\x4B\x00" }, /*ROLLBACK*/
        LDAP_OPT_ON /* -- critical ----- */
    };
static LDAPControl *hcdTC_rollback[3] = { &hcdTransactionControl_rollback,
                                        &hcdTransactionId, NULL };

```

Using these controls, the final request will, in the case of commit, now be:

```
rc = ldap_add_ext(ld, dn, pmods, hcdTC_commit, NULL, &msgidp);
```

and in the case of a rollback:

```
rc=ldap_add_ext(ld, dn, pmods, hcdTC_rollback, NULL, &msgidp);
```

If you look at the previous definition of hcdTC_commit and hcdTC_rollback, both parameters are control arrays that contain the commit or rollback control itself, and in addition, also contain the control hcdTC_Id for submitting the next request. Hence, one control indicates the end of the transaction, and the other control identifies the transaction on the basis of its ID. Once a transaction is closed, no further requests can be added to this control.

Appendix A. How to navigate through the dialog

This appendix illustrates the flow from the options on the HCD *Primary Task Selection* panel and the various actions that can be taken from each option. The panels that appear for option 1 provide an action bar at the top and a context menu for each object. Figure 161 on page 397 shows the general action bar valid for the action list panels. Some action lists offer special action bar choices that are not shown in the figure (for example the *Show/Hide* action bar choice on the Device List invoked from the Operating System List). Figure 155 on page 394 to Figure 159 on page 396 shows the options on the context menu and the navigation possibilities to other panels. Almost all of the options on the context menu can be directly selected by entering the action code next to the item in the list to be selected. Available action codes are shown in parentheses below the "Options Available" heading.

```

z/OS V2.1 HCD
Command ==> _____

Hardware Configuration

Select one of the following.

- 0. Edit profile options and policies
  1. Define, modify, or view configuration data
  2. Activate or process configuration data
  3. Print or compare configuration data
  4. Create or view graphical configuration report
  5. Migrate configuration data
  6. Maintain I/O definition files
  7. Query supported hardware and installed UIMs
  8. Getting started with this dialog
  9. What's new in this release

For options 1 to 5, specify the name of the IODF to be used.

I/O definition file . . . 'DOCU.IODF00.HCD.WORK'          +
  
```

Figure 152. *Primary Task Selection panel*

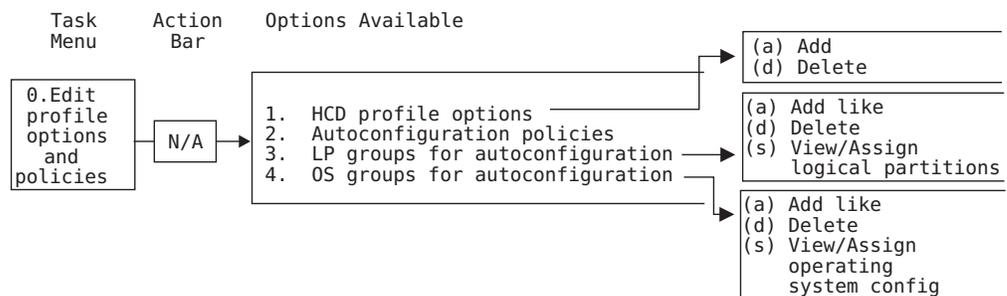


Figure 153. *HCD - Edit profile options and policies - Option 0*

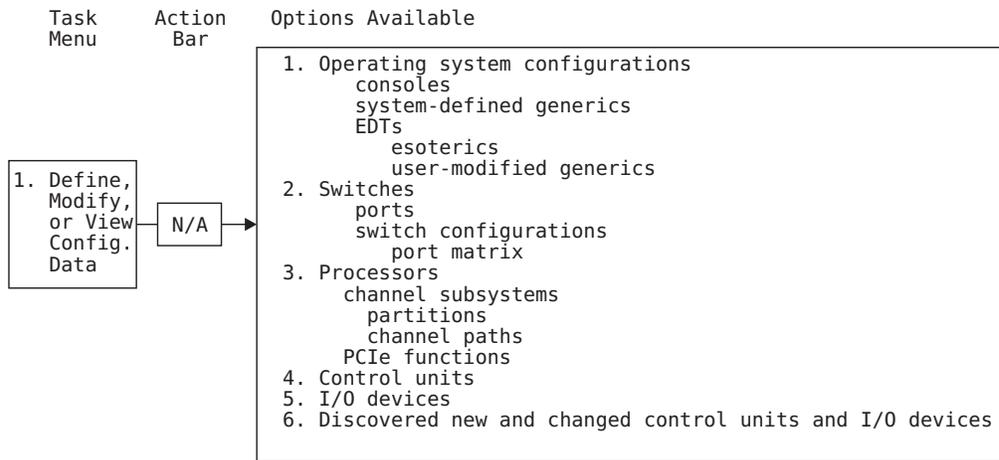


Figure 154. HCD - Define, Modify, or View Configuration Data - Option 1

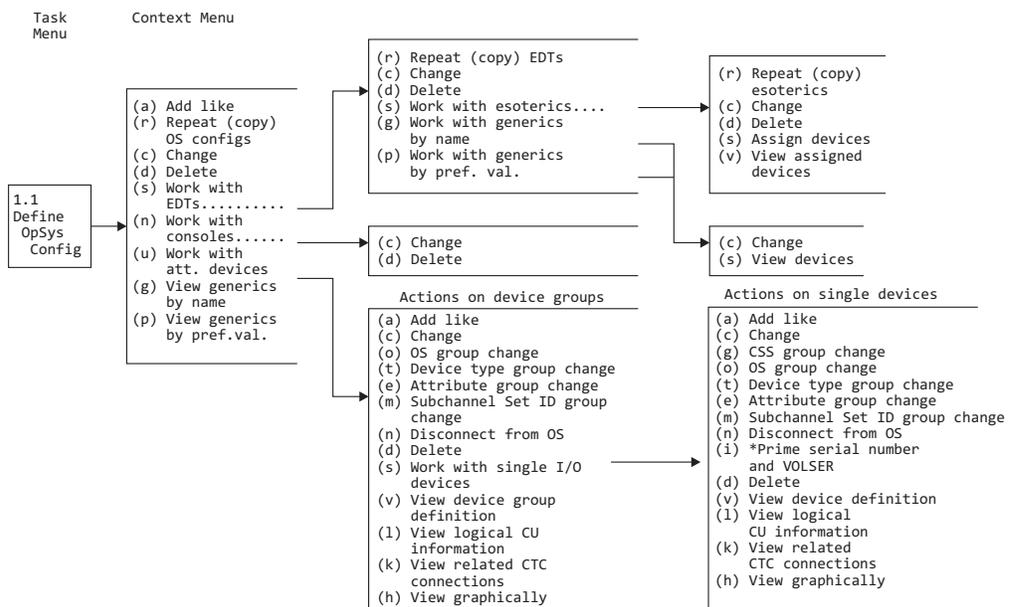


Figure 155. HCD - Define Operating System - Option 1.1

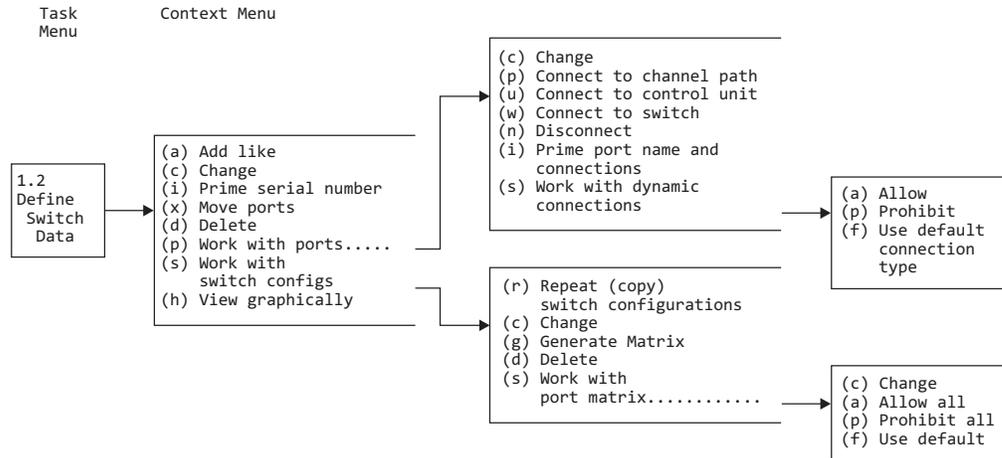


Figure 156. HCD - Define Switch - Option 1.2

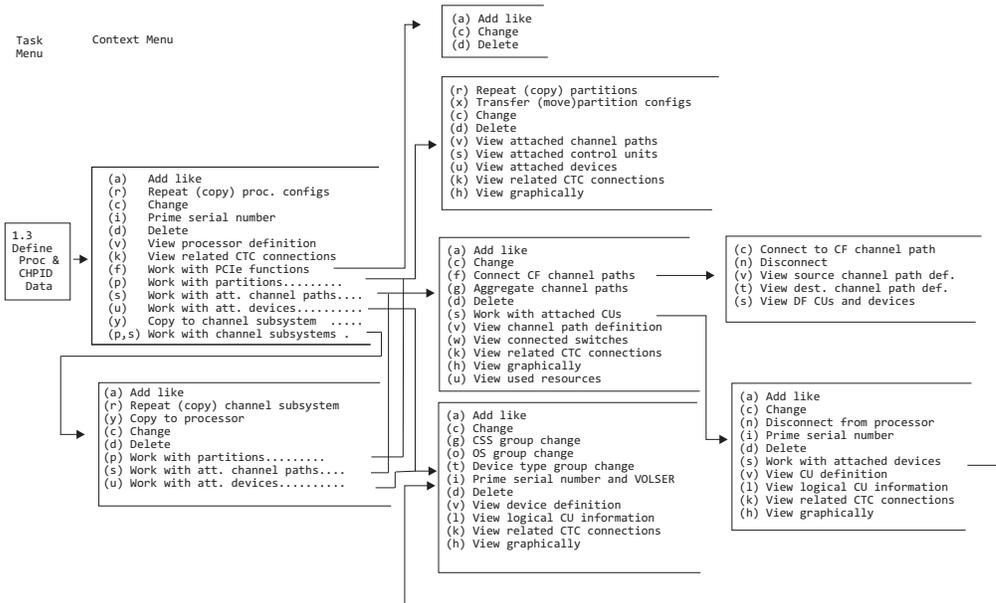


Figure 157. HCD - Define Processor and Channel Path - Option 1.3

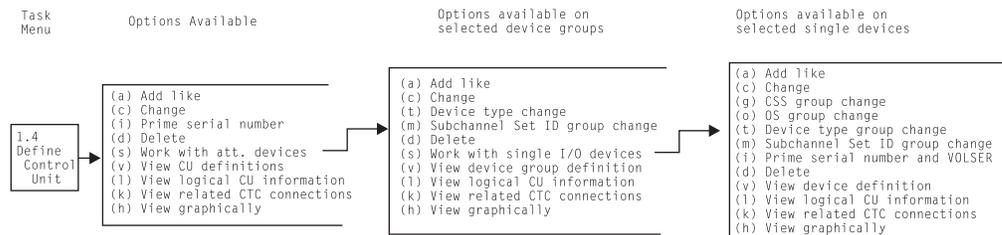


Figure 158. HCD - Define Control Unit - Option 1.4

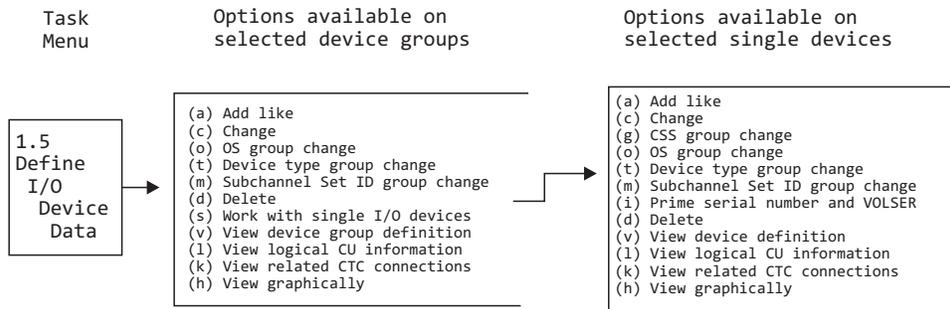


Figure 159. HCD - Define I/O Device - Option 1.5

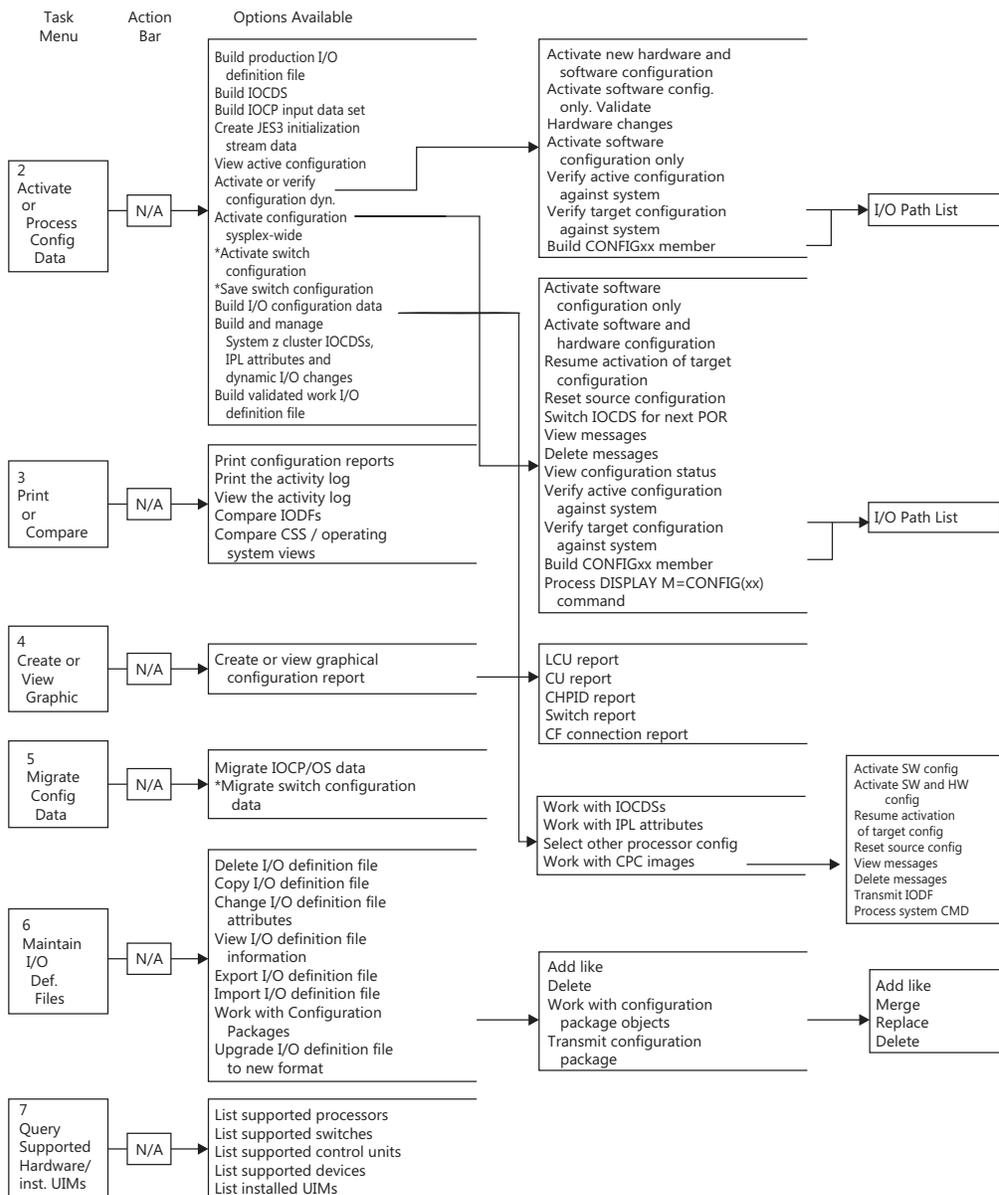


Figure 160. HCD - Primary Tasks - Options 2 - 7

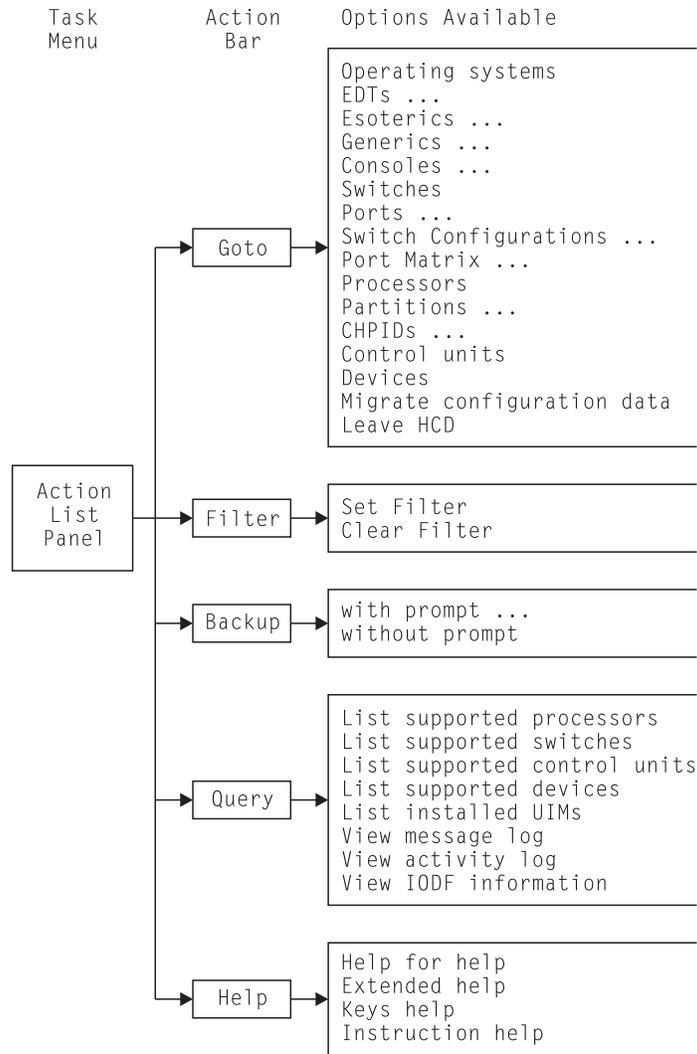


Figure 161. HCD - Generic Action Bar Options

Appendix B. Configuration reports

This information unit shows examples of the configuration reports that you can produce with HCD:

- “Textual configuration reports”
- “Graphical configuration reports” on page 454
- “IODF compare reports” on page 454.

Textual configuration reports

This section shows examples for textual configuration reports. “Print configuration reports” on page 253 lists the different report types available and describes how to produce them.

Channel subsystem reports

The following channel subsystem reports are available:

- “Processor Summary Report” on page 400
- “Channel Subsystem Summary Report” on page 400
- “PCIE function summary report” on page 400
- “CHID Summary Report” on page 402
- “Partition Report” on page 403
- “IOCDS Report” on page 403
- “Channel Path Summary Report” on page 404
- “Channel Path Detail Report” on page 406
- “CF Channel Path Connectivity Report” on page 407
- “Control Unit Summary Report” on page 409
- “Control Unit Detail Report” on page 409
- “Device Summary Report” on page 412
- “Device Detail Report” on page 412

Processor Summary Report

PROCESSOR SUMMARY REPORT						TIME: 15:29	DATE: 2012-10-21	PAGE A-	1
PROCESSOR ID	TYPE	MODEL	CONFIG. MODE	SERIAL NUMBER	DESCRIPTION	----- SNA ADDRESS -----		LSYSTEM	
						NETWORK NAME	CPC NAME		
ERV01WWW	2094	S38	LPAR	1212122094	mix system	ZZZZZZZ	ERV01	ERV01	
ERV0201	9672	ZX7	LPAR	4434349672	old system 1	IBM390PS	ERV0201		
ERV0202	9672	ZX7	LPAR	1111119672	old system 2	IBM390PS	ERV0202		
ERV05A1	2094	S28	LPAR	9912992094	Standalone FCTC CTC/CNC	IBM390PS	ERV05A1		
ERV05A2	2094	S54	LPAR	9929912094	standalone FCTC CTC/CNC proc	IBM390PS	ERV05A2		
NET01	2096	R07	LPAR	3322332096	Network System	ZZZZZZZ	NET01		

PROCESSOR ID	----SUPPORT LEVEL----	DESCRIPTION
ERV01WWW	H050331	XMP, Basic 2094 support
ERV0201	H990531	FCV, OSD, OSE, IC, ICB, CF, OSA, ESCON, Parallel
ERV0202	H990531	FCV, OSD, OSE, IC, ICB, CF, OSA, ESCON, Parallel
ERV05A1	H050331	XMP, Basic 2094 support
ERV05A2	H050331	XMP, Basic 2094 support
NET01	H060930	XMP, 2096-R07 support

Figure 162. Processor Summary Report

CONFIG. MODE	Indicates the operation mode in which a processor may operate. These modes are: BASIC The processor is not logically partitioned. LPAR The processor is logically partitioned. Several operating systems may run concurrently in different partitions of the processor.
SNA ADDRESS	The SNA Address consists of Network name and CPC name and associates the CPC and the processor definition in the IODF.
SUPPORT LEVEL	Shows the ID of the processor support level and an enumeration of the provided functionality.

Channel Subsystem Summary Report

Only XMP processors will have a channel subsystem report which shows the defined channel subsystems.

CHANNEL SUBSYSTEM SUMMARY REPORT								TIME: 12:55	DATE: 2014-06-17	PAGE A-	1
PROCESSOR ID	P2964	TYPE	2964	MODEL	N96						
CSS ID	DEVICES IN SS0	DEVICES IN SS1	DEVICES IN SS2	DEVICES IN SS3	DEVICES IN SS3	DEVICES IN SS3	DEVICES IN SS3	DESCRIPTION			
	MAXIMUM	ACTUAL	MAXIMUM	ACTUAL	MAXIMUM	ACTUAL	MAXIMUM	ACTUAL			
0	65280	16825	65535	0	65535	0	65535	0			
1	65280	16709	65535	0	65535	0	65535	0			
2	65280	14587	65535	0	65535	0	65535	0			
3	65280	15239	65535	0	65535	0	65535	0			
4	65280	14	65535	0	65535	0	65535	0	CSS 4 of P2964		
5	65280	0	65535	0	65535	0	65535	0			

Figure 163. Channel Subsystem Summary Report

Note: This report and the following channel subsystem reports show the processor token only if the IODF is a production IODF.

PCIe function summary report

The PCIe Function Summary Report provides an overview of the defined PCIe functions for the specified processor.

FID	VF	CHID	TYPE	UID	DESCRIPTION
0010	1	110	ROCE	F010	VF and 2 pnetids, a=lp01
0011	1	111	ROCE	F011	VF and 2 pnetids, a=lp01
0012	2	111	ROCE	F012	VF and 2 pnetids, a=lp01
0013	3	111	ROCE	F013	VF and 2 pnetids, a=lp01
001F	31	111	ROCE	F014	max VF and 2 pnetid,a=lp01
0020	31	4FE	ROCE	F020	max VF and 2 pnetid,a=lp01
0200	2	200	ZEDC-EXPRESS		Zedc 200, a=lp16
0222	3	200	ZEDC-EXPRESS		Zedc 200, a=lp16
0400	1	7C4	ISM		ism a=lp16
0401	2	7C4	ISM		ism a=lp16
0402	3	7C4	ISM		ism a=lp16
0403	4	7C4	ISM		ism a=lp16
0405	255	7C4	ISM		ism, max vf a=lp16
0FF0	1	4FF	ZEDC-EXPRESS		Zedc ff0,max pchid, a=lp01
1200		400	RCE		rce1200, no vf a=lp0a

PARTITION NUMBERS								
FID	VF	CHID	CSS0-----	CSS1-----	CSS2-----	CSS3-----	CSS4-----	CSS5-----
			123456789ABCDEF	123456789ABCDEF	123456789ABCDEF	123456789ABCDEF	123456789ABCDEF	123456789ABCDEF
0010	1	110	A-----	-----	CCCC-----	-----	-----	-----
0011	1	111	A-----	-----	CCCC-----	-----	-----	-----
0012	2	111	A-----	-----	CCCC-----	-----	-----	-----
0013	3	111	A-----	-----	CCCC-----	-----	-----	-----
001F	31	111	A-----	-----	CCCC-----	-----	-----	-----
0020	31	4FE	A-----	-----	CCCC-----	-----	-----	-----
0200	2	200	-CC-----	A-----	-----	-----	-----	-----
0222	3	200	-----	A-----	-----	-----	-----	-----
0400	1	7C4	-----	A-----	-----	-----	-----	-----
0401	2	7C4	-----	A-----	-----	-----	-----	-----
0402	3	7C4	-----	A-----	-----	-----	-----	-----
0403	4	7C4	-----	A-----	-----	-----	-----	-----
0405	255	7C4	-----	A-----	-----	-----	-----	-----
0FF0	1	4FF	ACCCC-----	-----	-----	-----	-C-----	-----
1200		400	---CC---ACC---	-----	-----	-----	-----	CCC-----

CSS0	CSS1	CSS2	CSS3	CSS4	CSS5
PARTITION	PARTITION	PARTITION	PARTITION	PARTITION	PARTITION
NUM NAME					
1 LP01	1 LP11	1 LP21	1 LP31	1 LP41	1 LP51
2 LP02	2 LP12	2 LP22	2 LP32	2 LP42	2 LP52
3 LP03	3 LP13	3 LP23	3 LP33	3 *	3 LP53
4 LP04	4 LP14	4 LP24	4 LP34	4 *	4 *
5 LP05	5 LP15	5 LP25	5 LP35	5 *	5 *
6 LP06	6 LP16	6 LP26	6 LP36	6 *	6 *
7 LP07	7 LP17	7 LP27	7 LP37	7 *	7 *
8 LP08	8 LP18	8 LP28	8 LP38	8 *	8 *
9 LP09	9 LP19	9 LP29	9 LP39	9 *	9 *
A LP0A	A LP1A	A LP2A	A LP3A	A *	A *
B LP0B	B LP1B	B LP2B	B *	B *	B *
C LP0C	C LP1C	C LP2C	C LP3C	C *	C *
D LP0D	D LP1D	D LP2D	D LP3D	D LP4D	D *
E LP0E	E LP1E	E LP2E	E LP3E	E LP4E	E *
F LP0F	F LP1F	F LP2F	F LP3F	F LP4F	F *

LEGEND FOR PARTITION NUMBERS FIELD:

- A - PARTITION IS IN FUNCTION ACCESS LIST
- C - PARTITION IS IN FUNCTION CANDIDATE LIST ONLY
- - PARTITION IS NOT IN FUNCTION ACCESS OR CANDIDATE LIST

Figure 164. PCIe Function Summary Report

CHID Summary Report

The *CHID Summary Report* provides an overview of all CHIDs defined for channel paths and PCIe functions of a processor and also gives an overview of the defined PNET IDs for the CHIDs.

CHID SUMMARY REPORT										TIME: 09:23 DATE: 2015-11-26 PAGE A- 3				
PROCESSOR ID		P2964V2		TYPE 2964		MODEL N30		CONFIGURATION MODE: LPAR						
CHID	VF	CHPID	FID	TYPE	SWITCH	CSS Numbers				PNET-1	PNET-2	PNET-3	PNET-4	
030			30	FC	17	0	1	2	3	4				
031			31	FC	17	0	1	2	3					
110	1	0010		ROCE		0		2			PNETID1	PNETID2		
1C4		A0		OSD					3					
320		A0		OSD		0								
391		50		FCP		0	1	2	3					
393		51		FCP		0	1	2	3					
3A1		A0		OSD				2						
3B0		86		OSD		0	1	2	3					
3C0		94		OSE		0								
400		2010		RCE		0		2						
401		2110		RCE		0		2						
4C2		C2		OSD		0	1	2		4	5			
4C3		C3		OSD		0	1	2		4	5			
4FE	30	0021		ROCE		0		2			PNETID1	ID2		
4FE	31	0020		ROCE		0		2			PNETID1	ID2		
4FF	5	0FF0		ZEDC-EXPRESS		0			3					
4FF	15	0FFF		ZEDC-EXPRESS		0			3	4	5			
7C0	10	1110		ISM		0					NET1			
7C0	11	1111		ISM		0					NET1			
7C0	12	1112		ISM		0					NET1			
7E0		F5		IQD		0	1	2	3	4	5			
7E5		FE		IQD		0	1	2	3					
*		76		OSD				2	3					
*		94		OSE				2						
*		94		OSE					3					
*		FA		IQD		0	1	2	3	4	5			
0F/2		10		CIB		0								
0F/2		11		CIB		0								
11/1		C5		CS5						4				
36/1		C4		CS5						4				

TOTALS FOR CHANNEL CARDS															
HCA		FICON		OSA				INTERNAL		PCIe					
CIB	CS5	FC	FCP	OSC	OSD	OSE	OSM	OSN	OSX	ICP	IQD	ISM	RCE	ROCE	ZEDC
1	2	2	2	0	7	3	0	0	0	0	3	1	2	2	1

CALCULATIONS OF TOTAL CHANNEL CARD VALUES:
HCA - A PHYSICAL ADAPTER (AID) WITH MULTIPLE PORTS IS COUNTED AS ONE ADAPTER
PCIe - A CHANNEL (CHID) WITH SEVERAL VIRTUAL FUNCTIONS (VF) IS COUNTED AS ONE ADAPTER

Figure 165. CHID Summary Report

Partition Report

PARTITION REPORT					TIME: 09:27	DATE: 2016-04-27	PAGE B-	1
PROCESSOR ID	P2964V3	TYPE	2964	MODEL	NC9			
CSS ID	0							
PARTITION NAME	NUMBER	USAGE	UID	DESCRIPTION				
LP0A	A	CF/OS	Y	Linux Test				
LP0B	B	OS	Y	Linux Test BTM				
LP0C	C	OS	Y	Linux Test				
LP0D	D	OS	N	Mini-VM 5 Guests				
LP0E	E	OS	N	Midi-VM 10 Guests				
LP0F	F	OS	N	Big-VM 60 Guests + BTM guests				
LP01	1	OS	Y	Test partition 01				
LP02	2	OS	N	Linux Test				
LP03	3	OS	N	Linux Test				
LP04	4	OS	N	Linux Test				
LP05	5	OS	N	Linux Test				
LP06	6	OS	N	Linux Test				
LP07	7	OS	N	Linux Test				
LP08	8	OS	N	Linux Test				
LP09	9	OS	N	Linux Test				

PARTITION REPORT					TIME: 09:27	DATE: 2016-04-27	PAGE B-	2
PROCESSOR ID	P2964V3	TYPE	2964	MODEL	NC9			
CSS ID	1							
PARTITION NAME	NUMBER	USAGE	UID	DESCRIPTION				
LP1A	A	OS	N	Linux Test				
LP1B	B	OS	N	Linux Test BTM				
LP1C	C	OS	N	Linux Test				
LP1D	D	OS	N	Mini-VM 5 Guests				
LP1E	E	OS	N	Midi-VM 10 Guests				
LP1F	F	OS	N	Big-VM 60 Guests				
LP11	1	OS	N	Linux Test				
LP12	2	OS	N	Linux Test				
LP13	3	OS	N	Linux Test				
LP14	4	OS	N	Linux Test				
LP15	5	OS	N	Linux Test				
LP16	6	OS	N	Linux Test				
LP17	7	OS	N	Linux Test				
LP18	8	OS	N	Linux Test				
LP19	9	OS	N	Linux Test				

Figure 166. Partition Report

NUMBER	Is the partition number (MIF ID). This information is printed only for EMIF capable processors.
USAGE	Specifies the usage type of a partition: CF indicates a partition supporting coupling facility. OS indicates a partition running an operating system. CF/OS indicates a partition supporting coupling facility or running an operating system.
UID	Specifies the partition flag if UID uniqueness is required and checked.

IOCDS Report

IOCDS data are retrieved from the support element when a SNA address is defined. Otherwise, the IOCDS data are retrieved from the IODF. An IOCDS status line at the end of the report indicates the source of the IOCDS data.

I O C D S R E P O R T										T I M E : 0 0 : 0 6 D A T E : 2 0 1 2 - 1 0 - 2 1 P A G E C - 1		
PROCESSOR ID	CFS	TYPE		9672	MODEL		RX6					
-----Token Match----- Write --Last Update--												
IOCDS NAME	FORMAT	STATUS	IOCDS/HSA	IOCDS/Proc	Protect	DATE	TIME	IOCDS	Configuration	Token	Information	
A0	316ACFS	LPAR	POR	Yes	No	Yes-POR	2004-11-12 06:14	CFS	16:27:52	04-11-11	SYS4	IODF71
A1	063ACFS	LPAR	Alternate	No	No	No	2004-03-03 16:49	CFS	16:23:49	04-03-03	SYS4	IODF72
A2	091ACFS	LPAR	Alternate	No	No	No	2004-04-01 00:01	CFS	16:08:02	04-03-31	SYS4	IODF73
A3	296ACFS	LPAR	Alternate	No	No	No	2004-10-22 11:45	CFS	14:59:05	04-10-21	SYS4	IODF71
IOCDS status retrieved from the support element												

Figure 167. IOCDS Report

NAME	Represents the user-defined name of the IOCDS (derived from the MSG1 parameter)
FORMAT	IOCDS format (BASIC or LPAR)
STATUS	Indicates the status of the IOCDS: <i>Alternate</i> , <i>POR</i> , <i>Invalid</i> (see "Build z Systems cluster IOCDSs" on page 215).
Token Match - IOCDS/HSA	Indicates whether the IOCDS token matches the current HSA token
Token Match - IOCDS/Proc	Indicates whether the IOCDS token matches the current processor token in the IODF
Write Protect	Indicates whether the IOCDS is write-protected (<i>Yes</i>) or not (<i>No</i>), or currently write-protected because it is the <i>POR IOCDS (Yes-POR)</i> .
Last Update DATE/TIME	Time stamp of IOCDS creation time
IOCDS Configuration Token Information	is the configuration token information stored in the support element and shows the relationship between the IOCDS and the production IODF from which it was created.

Channel Path Summary Report

If applicable, for spanned channels, there is a separate sub-report after the partition table of a processor which shows the connection of spanned channel paths to channel subsystems.

MFS KB	Designates the maximum frame size in KB. For channels of type IQD, this field also indicates whether or how this channel exploits the extended IQD function: blank indicates the HiperSockets function X indicates IEDN support (IQDX) B indicates bridge support
MNGD	Indicates whether the channel path is managed.
I/O CLUSTER	The I/O cluster name for managed channel paths.
DYN. SWITCH	Designates the switch holding the dynamic connection.
SWITCH ID	Designates the switch the channel is physically plugged in (entry switch).
PORT	Designates the entry port on the entry switch.
MODE	Operation mode of the channel path.

Channel Path Detail Report

The Channel Path Detail Report lists the channel paths defined per processor with their attributes and attachment information.

In addition, the switch connections on the path between the CHPID and the control unit are shown if they can be determined by HCD. The entry switch and port of the CHPID are always shown.

For an entry switch of a CHPID which is defined as a dynamic switch, the control unit port is shown if it is compatible with the link address defined for the CHPID. For an entry switch of the CHPID which is defined as a dedicated switch, the control unit port or the ports connecting the switches are only shown if switch configurations are defined which allow HCD determining a valid path between CHPID and control unit.

In case of chained switches, the first print line for a channel path shows the switch the channel path is physically plugged in. The second print line shows the switch the control unit is connected to. In addition, the first print line shows the ID of the switch with the dynamic connection.

Each attached control unit of a specific channel path is separated by a line.

The first print line for an attached control unit shows the first unit address range defined for the control unit. If there are more unit address ranges defined for a control unit, these are shown in the following print lines. Together with the unit address range(s) of the control unit the attached devices of the control unit are shown grouped according to device types and consecutive numbers and unit addresses. The report shows the starting device number and range of the device group. The unit address describes the address of the first device in the range.

For FICON switches, the dynamic switch ID is empty. The control unit port ID is shown as a two-byte port address (the port ID prefixed by the switch address) when used in a cascaded switch environment, or as a one-byte port address otherwise. For cascaded FICON switches, only the channel path port ID and the control unit port ID are shown, but no connection between the switches.

Note: For a Coupling Facility control unit, all CF devices attached to this control unit are listed, not only those devices that are defined for the connected coupling facility channel path described in the row.

CHANNEL PATH DETAIL REPORT											TIME: 12:57 DATE: 2012-10-21 PAGE E- 1					
PROCESSOR ID R35		TYPE 2817		MODEL M49		CONFIGURATION MODE: LPAR										
CSS ID 0																
---- SWITCH ----											UNIT ADDR		UNIT			
CHPID	AID/P	TYPE	MNGD	MODE	ID	PR	CU	DYN	--- CONTROL UNIT ---	CU-	PROTOCOL	RANGE	TO	-- DEVICE --	ADDR	DEVICE
						PN	PN	ID	NUMBER	TYPE-MODEL	ADD	FROM		NUMBER,RANGE	START	TYPE-MODEL
00	5B0	OSD	NO	SPAN					B302	OSA		00	FE	B300,6 B306,42 B330,16 B340,6 B346,42 B370,16	00 06 30 40 46 70	OSA OSA OSA OSA OSA OSA
0C	130	OSC	NO	SPAN					0003	OSC		00	FD	0090,16 1090,4	00 10	3270-X 3270-X
0D	131	OSM	NO	SPAN												
0E	1E1	CFP	NO	SPAN					FFDF	CFP				F0D4,7 F0DB,7		CFP CFP
0F	1E9	CFP	NO	SPAN					FFDF	CFP				F0D4,7 F0DB,7		CFP CFP
12	1F2	CNC	NO	SHR				1A	0690	3590		00	03	0690,4	00	3590
								1A	1A39	3590		00	0F	0A38,2	00	3590
								1A	1A55	3590		00	03	0A54,4	00	3590
30	220	FC	NO	SPAN	10	2A	6140		2000	2107	2	00	9F	2000,160	00	
					10	2A	611A		2800	2107	1	00	FF	2800,40 2828 2829,39 2850,2 2852,6 2858,88 28B0,5 28B5,11 28C0 28C1,8 28C9,7 28D0,48	00 28 29 50 52 58 B0 B5 C0 C1 C9 D0	3390B 3390B 3390B 3390B 3390B 3390B 3390B 3390B 3390B 3390B 3390A 3390B
					10	2A	612F		4001	2107	0	00	FF	4000,96 40A0,16	00 A0	3390B 3390B

Figure 169. Channel Path Detail Report

- SWITCH ID: Designates the entry switch the channel path is physically plugged in. For chained ESCON switches or cascaded FICON switches, a second line is shown with the ID of the switch to which the control unit is connected.
- SWITCH PR PN: Designates the entry port of the entry switch. In case of an ESCON chained switch, it designates the entry port of the chained switch.
- SWITCH CU PN: Designates the port the control unit is connected to. In case of an ESCON chained switch, it designates the port of the entry switch to which the chained switch is connected. For a cascaded FICON switch, the port ID is prefixed by the switch address.
- SWITCH DYN ID: For an ESCON environment, it designates the switch holding the dynamic connection.

CF Channel Path Connectivity Report

For XMP processors, there will be a single CF Channel Path Connectivity Report. The CHPID numbers are prefixed by the channel subsystem ID. For example 1.0C denotes CHPID 0C from CSS 1. If a spanned channel path is used, the CHPIDs are reported from all channel subsystems together with the accessed partitions in the channel subsystems.

PROCESSOR ID R35 TYPE 2817 MODEL M49 CONFIGURATION MODE: LPAR

SOURCE				DESTINATION				SOURCE		DESTINATION		CNTL		
CHPID	TYPE	MODE	ACCESS/CAND LIST	PROCESSOR ID	TYPE-MODEL	CHPID	TYPE	MODE	ACCESS/CAND LIST	CNTRL UNIT	DEVICE NUM,RANGE	CNTRL UNIT	DEVICE NUM,RANGE	TYPE
0.0E	CFP	SPAN	IRD7	H05	2097-E26	3.0E	CFP	SHR	*IRD7CF3	FFDF	F0D4,14			CFP
1.0E		N	IRD8											
0.0F	CFP	SPAN	IRD7	H05	2097-E26	3.0F	CFP	SHR	*IRD7CF3	FFDF	F0D4,14			CFP
1.0F		N	IRD8											
0.07	CFP	SPAN	IRD7	ECL2	2097-E40	0.81	CFP	SPAN	*IRD4CFF	FFD8	DFBA,7	FFD4	DFB3,7	CFP
1.07			(C)			3.81			D0E (C)		DFE4,7		DFDD,7	
			*IRD6						*D3F (C)					
0.08	CFP	SPAN	IRD7	H05	2097-E26	3.06	CFP	SHR	*DWB3CFF	FFD6	DF9E,7	FFD0	DF97,7	CFP
1.08			(C)								DFAC,7		DFA5,7	
2.08			*IRD6											
3.08			DWB1											
			DWB2											

LEGEND FOR ACCESS/CAND LIST:

- * - PARTITION IS OF TYPE CF OR CF/OS
- (C) - PARTITION IS IN CHPID'S CANDIDATE LIST ONLY

Figure 170. CF Channel Path Connectivity Report

SOURCE/DESTINATION CHPID	The identifier of the source/destination channel path.
SOURCE/DESTINATION TYPE	Is the type of the source/destination channel path.
SOURCE/DESTINATION MODE	Is the operation mode of the source/destination channel path.
SOURCE 0	CHPID is identified as occupied
SOURCE/DESTINATION ACCESS/CAND LIST	Shows those partitions which the source/destination channel path has in its access or candidate list. Partitions in the candidate list are flagged with (C) behind the partition name. Partitions of type CF or CF/OS are prefixed with an *. For example, *D3F (C) or *IRD6 (C) denote coupling facility partitions in the candidate list.
DESTINATION PROCESSOR ID	Is the name of the processor the destination channel path is defined for.
DESTINATION TYPE-MODEL	Is the type-model of the destination processor.
SOURCE CNTRL UNIT	Is the number of the CF control unit used for the source CF channel path connection.
SOURCE DEVICE NUM,RANGE	Are the numbers of the CF devices and ranges of device groups defined for the source CF channel path connections via the source CF control unit.
DESTINATION CNTRL UNIT	Is the number of the CF control unit used for the destination CF channel path connection.
DESTINATION DEVICE NUM,RANGE	Are the numbers of the CF devices and ranges of device groups used for the destination CF channel path connections via the destination CF control unit.
CNTL TYPE	Indicates the type of the connecting control unit(s).

Control Unit Summary Report

CONTROL UNIT		CONTROL UNIT SUMMARY REPORT		TIME: 16:12 DATE: 2012-10-21 PAGE G- 1				LOGICAL	
NUMBER	TYPE-MODEL	SERIAL NUMBER	DESCRIPTION	CONNECTED SWITCH ID.PORT NUMBER				PATHS	
0003	OSC		CU for OSC Channels						
0070	OSN								
0120	OSA		for all native MVS (also NETMVS)						
0480	3480	11333	TAPES 0480 CT1 BL6	1A.CB					
04B0	FCP		Open FCP 16						
0900	1750		DS6K1 (0900-094F)	80	17.54	17.58	17.5C	17.60	72
0950	1750		DS6K1 (0950-098F)	80	17.54	17.58	17.5C	17.60	72
0B20	2105	11111	ESS22 B20 CU2 (3390-3/27)VSE	152	17.45	17.49			34
0BC0	2105	11111	ESS22 BC0 CU4 (3390-mix) VSE	144	17.45	17.49			34
0C50	2105	11111	ESS22 C50 CU5 (3390-mix) VSE	176	17.45	17.49			34
0D80	2107	11222	DS8K0 CU0(D80-DCF) 3390-MIX	80	17.44	17.48	17.4C	17.50	424
0DD0	2107	11222	DS8K0 CU1(DD0-E1F) 3390-MIX	80	17.44	17.48	17.4C	17.50	424
0E20	2107	11222	DS8K0 CU2(E20-E3F) 3390-54	32	17.44	17.48	17.4C	17.50	424
0E40	2107	11222	DS8K0 CU3(E40-E5F) 3390-54	32	17.44	17.48	17.4C	17.50	0
0E60	2105	11111	ESS22 E60 CU0 (3390-9) VSE	96	17.45	17.49			0
0EC0	2105	11111	ESS22 EC0 CU1 (3390-3/27)VSE	152	17.45	17.49			0
0F60	2105	11111	ESS22 F60 CU3 (3390-65) VSE	64	17.45	17.49			0
1010	2032		McData ED-6064 z/OS						10.FE
1011	2032		McData ED-6140 z/OS						11.FE
1028	FCTC	H053E3F	FCTC H05LP04 to H05LP11 + TRX1						
102A	FCTC	H053E3F2	FCTC H05LP30 to H05LP31						
1128	FCTC	H053E3F	FCTC H05LP04 to H05LP11						
112A	FCTC	H053E3F2	FCTC H05LP30 to H05LP31						
112B	FCTC	H053E3F	FCTC H05LP04 to TRX1						
1240	OSA		OSA (1240-125E,5F)						

Figure 171. Control Unit Summary Report

CONNECTED SWITCH ID.PORT NUMBER Describes where the CU is physically connected to (switch and port)
 LOGICAL PATHS Displays the number of defined logical paths per control unit.

Control Unit Detail Report

The Control Unit Detail Report lists all control units defined in the currently accessed IODF, with their attributes and attachment information.

For each control unit all processors are shown, where the control unit is attached to. For each of those processors, all control units, that are part of the "logical control unit" the currently printed control unit belongs to, are printed. Also, the currently shown control unit is listed again to show information about switch connection, channel path attachment, and attached devices. The information about each logical control unit the currently printed control unit belongs to, is separated by a dotted line.

The switch connection information is shown in the Control Unit Detail Report even if no switch configuration is defined. For example:

If a control unit is connected to a switch, but no complete path is defined through the switch (that means, no valid path is defined through the switch, which allows a dynamic connection) the complete switch connection information with SWITCH ID, CU PN, and PR PN can be determined.

In case of chained switches, the first information in a print line is shown for the switch the control unit is connected to. The second set of switch information is for the switch that the channel path is connected to. Note that this is different from the Channel Path Detail report.

To get information about the switch with the dynamic connection (in case of chained ESCON switches), the Channel Path Detail Report has to be produced.

For each control unit belonging to a logical control unit, the channel path it is attached to, is printed together with the link address (if applicable).

For FICON cascade switching, the link address will be shown as a two-byte number. If a path from the control unit is via cascaded FICON switches, the link between the FICON switches is not shown. Instead, the control unit port of the switch connected to the control unit and the CHPID port of the switch connected to the channel path are shown. If no port connections are defined, switch data is extracted from the dynamic switch and the link address. Port IDs are shown as one-byte port addresses.

The 'DEVICE' column shows the devices which are attached to the control unit printed under column 'CU IN LCU'. If the control unit is attached to more than one channel path, the information about the attached devices is printed together with the last printed channel path. The devices attached to a control unit are grouped according to consecutive number. The report shows the starting device number and range of the device group.

Columns 'LOG. PATHS PER CU' and 'LOGICAL PATHS PER CU PORT' display the number of defined logical paths.

CONTROL UNIT DETAIL REPORT													TIME: 16:12	DATE: 2012-10-21	PAGE H-	2
CONTROL UNIT NUMBER	TYPE-MODEL	PROCESSOR.CSS ID	LOG. PATHS PER CU	CU IN LCU	IOCL	CU-ADD	SWITCH ID	CU PN	PR PN	-CHAINED/CASC-SWITCH ID	CU PN	PR PN	CHPID LINK ADDR	-- DEVICE NUMBER,RANGE	LOGICAL PATHS PER CU PORT	
0950	1750	PROC03.1	4	0950		3	17 17	5C 60		14 14	52 4E	35 41	. 685C . 6860	0950,80	20 20	
		PROC03.2	4	0950		3	17 17	5C 60		14 14	52 4E	35 41	. 685C . 6860	0950,80	20 20	
0B20	2105	PROC03.0	2	0B20		2	17 17	45 49		14 14	52 4E	35 41	. 6845 . 6849	0B20,152	27 27	
		PROC03.1	2	0B20		2	17 17	45 49		14 14	52 4E	35 41	. 6845 . 6849	0B20,152	27 27	
		PROC03.2	4	0B20		2	17 17	45 49		14 14	52 4E	35 41	. 6845 . 6849	0B20,152	27 27	
		PROC03.3	2	0B20		2	17 17	45 49		14 14	52 4E	35 41	. 6845 . 6849	0B20,152	27 27	
0BC0	2105	PROC03.0	2	0BC0		4	17 17	45 49		14 14	52 4E	35 41	. 6845 . 6849	0BC0,144	18 18	
		PROC03.1	2	0BC0		4	17 17	45 49		14 14	52 4E	35 41	. 6845 . 6849	0BC0,144	18 18	
		PROC03.2	4	0BC0		4	17 17	45 49		14 14	52 4E	35 41	. 6845 . 6849	0BC0,144	18 18	
		PROC03.3	2	0BC0		4	17 17	45 49		14 14	52 4E	35 41	. 6845 . 6849	0BC0,144	18 18	
0C50	2105	PROC03.0	2	0C50		5	17 17	45 49		14 14	52 4E	35 41	. 6845 . 6849	0C50,170	9 9	
		PROC03.1	2	0C50		5	17 17	45 49		14 14	52 4E	35 41	. 6845 . 6849	0C50,170	9 9	
		PROC03.2	4	0C50		5	17 17	45 49		14 14	52 4E	35 41	. 6845 . 6849	0C50,170	9 9	
		PROC03.3	2	0C50		5	17 17	45 49		14 14	52 4E	35 41	. 6845 . 6849	0C50,170	9 9	

Figure 172. Control Unit Detail Report

Table 29. Description of the columns in the Control Unit Detail Report

Column name	Description
PROCESSOR.CSS ID	Designates the processors, and in case of an XMP processor, the channel subsystem to which the CU is attached.
LOG. PATHS PER CU	Specifies the number of defined logical paths for a control unit per channel subsystem.
CU IN LCU	Designates which CUs belong to the logical CU.
IOCL	Designates I/O concurrency level (same as SHARED in IOCP). Specifies the level of concurrency of I/O requests that the parallel channel path allows for the control unit (CU). 1 One I/O request at a time. (SHARED=Y) 2 Multiple I/O requests at a time. (SHARED=N)
CU-ADD	Designates the CU address.
SWITCH ID	Designates the switch the CU is connected to.
CU PN	Designates the port the CU is physically connected to.
PR PN	Designates the entry port of the channel (except in the case of chained switches)
CHAINED/CASC (SWITCH ID, CU PN, PR PN)	Designates a possible second switch the CHPID is connected to.

Table 29. Description of the columns in the Control Unit Detail Report (continued)

Column name	Description
CHPID . LINK ADDR	Designates the channel path and the one- or two-byte link address to which the control unit is connected. <i>n</i> (*) in this column indicates that the control unit is connected to <i>n</i> managed channel paths.
DEVICE NUMBER,RANGE	Specifies information about the devices that are attached to the processors by the control unit.
LOGICAL PATHS PER CU PORT	Specifies the number of defined logical paths for all control units connected to a specific port.

Device Summary Report

The Device Summary Report gives you an overview of the devices defined in the currently accessed IODF and their attaching control units. The devices are grouped according to the same characteristics. The report shows the starting device number and the range of the group.

For multi-exposure devices, the base devices are grouped separately from the non-base exposure devices.

DEVICE SUMMARY REPORT			TIME: 16:04	DATE: 2012-10-21	PAGE I-	2	PPRC
--- DEVICE --- NUMBER,RANGE	DEVICE TYPE-MODEL	ATTACHING CONTROL UNITS	SERIAL NUMBER	DESCRIPTION	VOLSER	USE	
0080,16	3174	0080		Merian SNA Connection			
0090,2	RS6K	0110					
0092,2	RS6K	0112					
0094,2	RS6K	0114					
0096,2	RS6K	0116					
01C0	3270-X	01C0		Console Real and Virt.			
01C1,31	3270-X	01C0		Terminal			
2100,25	3390A	2124					
2200,25	3390B	2126					U
2300,25	3390B	2126					D
2400,25	3390B	2128					D
0300,32	9345	300A 300B					
0320,32	9345	320A 320B					
0340,32	9345	340A 340B		Real Dasd (9345 340-35f)	MVSLIB		
0360,32	9345	360A 360B		Real Dasd (9345 360-37f)	TS0001		

Figure 173. Device Summary Report

Device Detail Report

The Device Detail Report lists all devices defined in the currently accessed IODF, with their attributes and attachment information. Each device is shown with the processors to which it is attached. The device - processor attachment attributes are also listed. If applicable, the subchannel set where the device is located, is also shown in column SS. If not applicable, this column shows a blank.

For each processor the device is attached to, the CUs for the attachment to the processor, as well as the channel path(s) the CU is attached to, are also listed.

The starting device number and the range of subsequent device numbers are shown in one row. A range value of 1 is omitted.

The Device Detail Report shows a partition matrix which indicates whether a logical partition of a corresponding processor has access to the device either via the channel path access list or the channel path candidate list, and whether a partition is excluded or included via the device candidate list. Devices with a null-device candidate list are excluded from the report.

At the end of the report, for SMP processors, the totals for the following items are listed for each processor:

- CHPIDS
- PHYSICAL CONTROL UNITS
- SUBCHANNELS
- LOGICAL CONTROL UNITS

For the CHPID total, the report lists separate values for the total shared and the total unshared. For the physical CU total, HCD reports the total shared (those attaching to shared channel paths) and the total unshared.

For the subchannel and logical CU totals, the report lists separate values for the shared, unshared, and additional unshared counts that are, respectively, assigned and unassigned to a logical partition. The shared count is the total number assigned to shared channel paths. The generated for LPAR unshared count is the total that would have been generated for a basic IOCDs. The additional unshared count contains the unshared values that were generated for the LPAR IOCDs.

The TOTAL value is the total that would have been contained in the generated IOCDs. The HSA TOTAL is the total that will exist in the HSA after POR. At the completion of POR, the HSA may contain more subchannels and logical CUs than does the IOCDs.

For XMP processors, the following totals are reported:

- CHPIDS
- PHYSICAL CONTROL UNITS
- DEVICES
- LOGICAL CONTROL UNITS

The column CSS TOTAL lists the number of CHPIDs, physical and logical control units and the maximum number of devices that are currently defined for that channel subsystem.

The column IOCDs TOTAL lists the number of CHPIDs, physical and logical control units and the maximum number of devices without definitions caused by over-defined CHPIDs.

Columns HSA TOTAL and HSA LIMIT are not applicable for XMP processors.

The column USER LIMIT lists the maximum number of devices defined by the user for that channel subsystem.

The column SUPPORTED LIMIT lists the maximum number of CHPIDs, physical and logical control units and the maximum number of devices that are supported for the processor for that channel subsystem.

DEVICE DETAIL REPORT										TIME: 13:33	DATE: 2012-10-21	PAGE J-	1														
DEVICE NUMBER	DEVICE TYPE-MODEL	PROCESSOR. CSS ID	SS ADDR	UNIT OUT	TIME STA DET	PREFERRED CHPID	CNTL PORT NUMBER	UNIT CUADD	CHPID. LINK	PARTITION NUMBERS										SIDE							
										1 2 3 4 5 6 7 8 9 A B C D E F																	
0000,8	3390A	GOLDENE1.0	0	00	NO	YES	0000	00	01	A	A																
		GOLDENE1.1	0	00	NO	YES	0000	00	02	A	A																
		GOLDENE1.2	0	00	NO	YES	0000	00	02	A	A																
		GOLDENE1.3	0	00	NO	YES	0000	00	01																		
		GOLDENE1.3	0	00	NO	YES	0000	00	00	-	-																
...																											
0020,16	3380A	GOLDENE1.0	1	30	NO	YES	0000	00	01	A	A																
		GOLDENE1.1	1	30	NO	YES	0000	00	02	A	A																
		GOLDENE1.2	1	30	NO	YES	0000	00	02	A	A																
		GOLDENE1.3	1	30	NO	YES	0000	00	01																		
		GOLDENE1.3	1	30	NO	YES	0000	00	00	-	-																

DEVICE DETAIL REPORT TIME: 13:33 DATE: 2012-10-21 PAGE J- 2

LEGEND FOR PARTITION NUMBERS FIELD:

- A - PARTITION IS IN CHPID'S ACCESS LIST
- C - PARTITION IS IN CHPID'S CANDIDATE LIST ONLY
- BLANK - PARTITION IS NOT IN CHPID'S ACCESS OR CANDIDATE LIST
- - PARTITION IS IN CHPID'S ACCESS OR CANDIDATE LIST BUT PARTITION IS EXCLUDED FROM DEVICE'S CANDIDATE LIST
- + - PARTITION IS NOT IN CHPID'S ACCESS OR CANDIDATE LIST BUT PARTITION IS INCLUDED IN DEVICE'S CANDIDATE LIST

DEVICE DETAIL REPORT TIME: 13:33 DATE: 2012-10-21 PAGE J- 3

PROCESSOR: GOLDENE1
 CSS ID: 0

PARTITION NUMBER	NAME
1	GECSS01X
3	GECSS03X
9	GECSS09X
F	GECSS0FX

TOTALS FOR CHPIDS, DEVICES AND CONTROL UNITS

	CSS TOTAL	IOCDS TOTAL	USER LIMIT	SUPPORTED LIMIT
CHPIDS	3	3	N/A	256
PHYSICAL CONTROL UNITS	1	1	N/A	8192
DEVICES IN SUBCHANNEL SET 1	32	32	1000	64512
DEVICES IN SUBCHANNEL SET 2	32	32	300	65535
LOGICAL CONTROL UNITS	1	1	N/A	4096

Figure 174. Device Detail Report

Switch reports

- The following switch reports are available:
- "Switch Summary Report" on page 415
 - "Switch Detail Report" on page 415
 - "Switch Configuration Summary Report" on page 415
 - "Switch Configuration Detail Report" on page 416

Switch Summary Report

SWITCH SUMMARY REPORT					TIME: 14:54	DATE: 2012-10-21	PAGE K- 1
SWITCH ID	TYPE	ADDR	SERIAL NUMBER	DESCRIPTION	CU NUMBER	DEVICE NUMBER	INSTALLED PORTS
21	2032	41		switch address 41 fabric a			33
98	9033		55-9999	First switch	0098	0098	9
99	9032		55-8888	Second switch	0099	0099	29

Figure 175. Switch Summary Report

ADDR Shows the switch address, if available.
 CU NUMBER Shows all switch control units attached to the switch CU port of the switch.
 DEVICE NUMBER Shows all switch devices defined for a switch.

Switch Detail Report

If the switch is connected to an XMP processor, the processor ID is qualified with the channel subsystem ID. A spanned channel path is suffixed with an asterisk (*).

SWITCH DETAIL REPORT							TIME: 10:53	DATE: 2012-10-21	PAGE L- 3
PORT	PORT NAME	UNIT	UNIT ID	CONNECTION	UNIT TYPE	OCCUPIED			
AE		CU	053E		3490	NO			
AF	CHAIN_FROM_99	SW	99	PORT F0	9033	NO			
B0	CU_400	CU	0400		3190	NO			
B1	PROCA_CP18					YES			
B2	PROCA_CP19					YES			
B3		CU	0900		3990	NO			
B4		PR	G29.1	CHPID 32	2084-C24	NO			
		CU	1060		SCTC				

Figure 176. Switch Detail Report

CONNECTION UNIT Specifies the type of the unit the port is connected to.
 PR = Processor, CU = Control Unit, SW = Switch
 OCCUPIED Indicates a port connection external to the IODF.

Switch Configuration Summary Report

SWITCH CONFIGURATION SUMMARY REPORT					TIME: 17:24	DATE: 2012-10-21	PAGE M- 1
SWITCH ID	SWITCH TYPE	CONFIG ID	DEFAULT CONNECTION	CONFIGURATION DESCRIPTION			
98	9033	BASIC	ALLOW	night-shift configuration			
99	9032	BASIC2	ALLOW				

Figure 177. Switch Configuration Summary Report

DEFAULT CONNECTION Indicates what state a potential dynamic connection may have (allow or prohibit).

Switch Configuration Detail Report

The Switch Configuration Detail Report lists all supported ports of a switch with their dynamic connection attributes.

SWITCH CONFIGURATION DETAIL REPORT										TIME: 14:59	DATE: 2012-10-21	PAGE N-	1		
SWITCH ID	SWITCH TYPE	CONFIGURATION ID	DEFAULT CONNECTION	INSTALLED PORTS											
00	9032-5	SW00ID00	ALLOW	29											
PORT	PORT NAME	BLOCKED	DEDICATED	ALLOWED CONNECTIONS	PROHIBITED CONNECTIONS										
04	PR EVA.0	CHP 00 2084	NO	05-0A,0C-1F	0B										
05	PR EVA.0	CHP 01 2084	NO	04,06-1F											
06	CU 0000	2105	NO	04-05,07-1F											
07	PR EVA.0	CHP 08	NO	10	04-06,08-1F										
08	PR R31SMP	CHP 00 2066	NO	0F	04-07,09-1F										
09	PR EVA.0	CHP 09	NO	0F	04-08,0A-1F										
0A	PR EVA.0	CHP 10 2084	NO		04-09,0B-1F										
0B	SW 01	PO 04 9032	NO		05-0A,0C-1F	04									
0C	PR EVA.0	CHP 02 2084	NO		04-0B,0D-1F										
0D	CU 0003	3490	NO		04-0C,0E-1F										
0E	CU 0005	2105	NO		04-0D,0F-1F										
0F	SW 01	PO 05	NO	09	04-0E,10-1F										
10	SW 01	PO 06	NO	07	04-0F,11-1F										
11			NO		04-10,12-1F										
12			NO		04-11,13-1F										

Figure 178. Switch Configuration Detail Report

Operating System reports

The following operating system reports are available:

- “Operating System Summary Report”
- “MVS Device Report”
- “MVS Device Detail Report” on page 418
- “Eligible Device Table Report” on page 419
- “NIP Console Report” on page 420
- “VM Device Report” on page 420
- “VM Device Detail Report” on page 421
- “VM Console Report” on page 421

Operating System Summary Report

OPERATING SYSTEM SUMMARY REPORT					TIME: 13:37	DATE: 2012-10-21	PAGE 0-	1
OPERATING SYSTEM ID	TYPE	GEN	DESCRIPTION	D/R SITE OS ID				
OPSYS01	MVS		MVS operating system	OPSYS01D				
OPSYS02	VM		VM operating system					
OPSYS01D	MVS	YES	generated for OPSYS01					

Figure 179. Operating System Summary Report

MVS Device Report

The MVS Device Report gives an overview of the devices defined to an MVS-type operating system in the currently accessed IODF.

The devices are grouped according to same characteristics. The report shows the starting device number and the range of a group.

MVS DEVICE REPORT										TIME: 13:27		DATE: 2012-10-21		PAGE P- 1	
OPERATING SYSTEM CONFIGURATION ID: OS000001															
DEV#,RANGE	TYPE-MODEL	SS	BASE	UCB-TYPE	ERP-NAME	DDT-NAME	MLT-NAME	OPT	UIM-NAME	ATI	AL	SH	SW	MX	MI
0000,16	3390A	0		3010200F	IECVDERP	IGGDDTA1	IEAMLT02		CBDUS002	00			Y		
0010,16	3390A	1		3010200F	IECVDERP	IGGDDTA1	IEAMLT02		CBDUS002	00			Y		
0020,16	3380A	0		3010200E	IECVDERP	IGGDDT01	IEAMLT02		CBDUS002	00			Y		
0020,16	3380A	1		3010200E	IECVDERP	IGGDDT01	IEAMLT02		CBDUS002	00			Y		

MVS DEVICE REPORT										TIME: 13:27		DATE: 2012-10-21		PAGE P- 2	
KEY	KEY DESCRIPTION														
---	-----														
DEV#,RANGE	- DEVICE NUMBER, COUNT OF DEVICES (DECIMAL)														
TYPE-MODEL	- DEVICE NAME														
SS	- SUBCHANNEL SET ID														
BASE	- BASE DEVICE NUMBER FOR MULTIPLE EXPOSURE DEVICES														
UCB-TYPE	- UCB TYPE BYTES														
ERP-NAME	- ERROR RECOVERY PROGRAM														
DDT-NAME	- DEVICE DESCRIPTOR TABLE														
MLT-NAME	- MODULE LIST TABLE														
OPT	- OPTIONAL MLT INDICATOR														
UIM-NAME	- UNIT INFORMATION MODULE SUPPORTING THE DEVICE														
ATI	- ATTENTION TABLE INDEX (UCBATI)														
AL	- ALTERNATE CONTROL UNIT (UCBALTCU)														
SH	- SHARED UP OPTION (UCBSHRUP)														
SW	- DEVICE CAN BE SWAPPED BY DDR (UCBSWAPP)														
MX	- DEVICE HAS MULTIPLE EXPOSURES (UCBMTPXP)														
MI	- MIH PROCESSING SHOULD BE BYPASSED (UCBMIHPB)														
O	- MLT IS OPTIONAL														
Y	- DEVICE SUPPORTS THIS FEATURE														
BLANK	- DEVICE DOES NOT SUPPORT THIS FEATURE														

TOTAL NUMBER OF DEVICES BY CLASS		

CLASS NAME	CLASS TYPE	DEVICE COUNT
-----	-----	-----
TAPE	80	0
COMMUNICATION DEVICES	40	0
C-T-C	41	0
DASD	20	64
GRAPHICS	10	0
UNIT RECORD	08	0
CHARACTER READERS	04	0
TOTAL NUMBER OF I/O DEVICES DEFINED BY THIS I/O CONFIGURATION		64

Figure 180. MVS Device Report

MVS Device Detail Report

OPERATING SYSTEM	CONFIGURATION ID:	MVS DEVICE DETAIL REPORT		TIME: 15:11	DATE: 2012-10-21	PAGE Q-	1
--- DEVICE ---	DEVICE						
NUMBER,RANGE	TYPE - MODEL	SS	PARAMETER	FEATURE			
0000,16	3390A	0	WLMPAV=YES				
0010,16	3390A	1	WLMPAV=YES				
0020,16	3380A	1	WLMPAV=YES				
0020,16	3380A	0	WLMPAV=YES				
0040,8	3380A	1	WLMPAV=NO				
0048,8	3380A	1	WLMPAV=YES				
0100,8	3390B		OFFLINE=NO,DYNAMIC=YES,LOCANY=YES, WLMPAV=YES	SHARED			
FFFE	3179		OFFLINE=NO,DYNAMIC=YES,LOCANY=YES	DOCHAR			
FFFF	3179		OFFLINE=YES,DYNAMIC=YES,LOCANY=YES	DOCHAR			

Figure 181. MVS Device Detail Report

PARAMETER Shows the parameter values specified for the devices. If you do not specify "Yes" or "No" for devices that support the dynamic capability, the DYNAMIC parameter will not be displayed.

FEATURE Shows the features given to these devices.

Eligible Device Table Report

E D T R E P O R T							TIME: 14:40 DATE: 2012-10-21 PAGE R- 1	
OPERATING SYSTEM CONFIGURATION ID: OPSYS01								
EDT IDENTIFIER: A1 DESCRIPTION: special								
NAME	NAME TYPE	VIO	TOKEN	PREF	AFFINITY INDEX	ALLOCATION DEVICE TYPE	ASSOCIATED GENERICS	DEVICE NUMBER LIST
3390	GENERIC			280	FFFF	3010200F		01D1-01D8
3480	GENERIC			1100	0008	78008080		00C1
3277-2	GENERIC			3800	FFFF	12001009		0001
SWCH	GENERIC			0500	FFFF	0000083A		0098
ES001	ESOTERIC			3800	FFFF	12001009	3277-2	0001
ES002	ESOTERIC			280	FFFF	3010200F	3390	01D1-01D8
				1100	0008	78008080	3480	00C1
3400-9	G/GENERIC			1100	0008	78008080	3480	00C1
SYSALLDA	G/ESOTERIC			280	FFFF	3010200F	3390	01D1-01D8
SYS3480R	G/ESOTERIC			1100	0008	78008080	3480	00C1

Figure 182. Eligible Device Table Report

NAME TYPE	describes the type of the device groups contained in the EDT:
	<p>GENERIC generic device type group</p> <p>ESOTERIC esoteric device group</p> <p>G/GENERIC system generated generic device type group</p> <p>G/ESOTERIC system generated esoteric device group</p>
VIO	Eligible for virtual I/O, designates temporary data sets that exist in paging storage only.
PREF	Preference value, indicates the order the system should follow when attempting allocation.
AFFINITY INDEX	<p>This index is used by the system allocation programs to determine which devices have affinity to each other. Devices have affinity if either of the following statements are true:</p> <ul style="list-style-type: none"> • The devices have the same affinity index. • The affinity index for one of the devices is a subset of the other devices' affinity index. <p>An affinity index is a subset of another if both of the following statements are true:</p> <ul style="list-style-type: none"> • Neither index has a value of X'FFFF'. • One or more bits in one index are set to a binary "one" and one or more corresponding bits in the other index are set to a binary "one".
ALLOCATION DEVICE TYPE	UCB device table for allocation entry.
ASSOCIATED GENERICS	Indicates the relation of a device type to generics.
DEVICE NUMBER LISTS	<p>Lists the devices that are included in the group. The devices are grouped according to subsequent device numbers. The report shows the range of the device group.</p> <p>Note: Device ranges in a subchannel set with a subchannel set ID > 0 are displayed in a 5-digit notation with the leading digit indicating the subchannel set ID. For example, a device range 1000-103F located in subchannel set 1 is shown as 11000-1103F. A device range 2000-203F in subchannel set 0 is shown as 2000-203F.</p>

NIP Console Report

```

                                N I P C O N   R E P O R T
OPERATING SYSTEM CONFIGURATION ID: OPSYS01          TIME: 13:37  DATE: 2012-10-21  PAGE S-  1
                                NIP CONSOLE DEVICES
                                -----
                                DEVICE #      TYPE-MODEL
                                -----
                                0001          3278-3
  
```

Figure 183. NIP Console Report

VM Device Report

The VM Device Report gives an overview of the devices defined to a VM operating system in the currently accessed IODF.

The devices are grouped according to same characteristics. The report shows the starting device number and the range of a group.

```

                                VM DEVICE REPORT
OPERATING SYSTEM CONFIGURATION ID: VMXXL001          TIME: 13:48  DATE: 2012-10-21  PAGE P-  1
DEV#,RANGE  TYPE-MODEL  SS      BASE      CLASS      VIRT  UIM-NAME  MX  DO  US
-----
0002        3279-2      -        -        -        -        -
01D1,4      3390
0020,16     3380A          0        DASD      CBDUS258
  
```

```

                                VM DEVICE REPORT
                                TIME: 13:48  DATE: 2012-10-21  PAGE P-  2
KEY          KEY DESCRIPTION
---          -----
DEV#,RANGE  - DEVICE NUMBER, COUNT OF DEVICES (DECIMAL)
TYPE-MODEL  - DEVICE TYPE AND MODEL
SS          - SUBCHANNEL SET ID
BASE        - BASE DEVICE NUMBER FOR MULTIPLE EXPOSURE DEVICES
CLASS       - VM DEVICE CLASS
VIRT        - DEVICE IS NOT DEFINED TO CHANNEL SUBSYSTEM
UIM-NAME    - UNIT INFORMATION MODULE SUPPORTING THE DEVICE
MX          - DEVICE HAS MULTIPLE EXPOSURES
DO          - DEVICE IS SUPPORTED DEDICATED-ONLY
US          - DEVICE IS UNSUPPORTED
Y           - DEVICE SUPPORTS THIS FEATURE
BLANK       - DEVICE DOES NOT SUPPORT THIS FEATURE
  
```

```

TOTAL NUMBER OF DEVICES BY CLASS
-----
CLASS NAME      DEVICE COUNT
-----
TERMINAL        0
GRAPHIC         0
REMOTE GRAPHIC  0
SPOOL           0
TAPE            0
DASD            16
SPECIAL         0
TOTAL NUMBER OF I/O DEVICES DEFINED BY THIS I/O CONFIGURATION 16
  
```

Figure 184. VM Device Report

VM Device Detail Report

OPERATING SYSTEM CONFIGURATION ID: OPSYS02		VM DEVICE DETAIL REPORT		TIME: 15:06	DATE: 2012-10-21	PAGE Q-	1
--- DEVICE --- NUMBER,RANGE	DEVICE TYPE - MODEL	SS	PARAMETER	FEATURE			
0000,16	3390A	0	WLMPAV=YES				
0100	3278-3		OFFLINE=NO,OFFLINE=NO				
0200	3279-2		OFFLINE=NO,OFFLINE=NO				
02D1,4	3390		OFFLINE=NO,OFFLINE=NO,OFFLINE=NO				

Figure 185. VM Device Detail Report

PARAMETER Shows the parameter values specified for the devices.
FEATURE Shows the features given to these devices.

VM Console Report

OPERATING SYSTEM CONFIGURATION ID: OPSYS02		VM CONSOLE REPORT		TIME: 13:45	DATE: 2012-10-21	PAGE S-	1
		VM CONSOLE DEVICES					
DEVICE #	TYPE-MODEL						
0002	3279-2	PRIMARY SYSTEM CONSOLE					

Figure 186. VM Console Report

CTC Connection Report

CTC CONNECTION REPORT														TIME: 14:28 DATE: 2012-10-21 PAGE T- 1													
LINE	SIDE 1							COMMON							SIDE 2												
	PROC.CSSID	PART. NAME	DEVICE NUM	OS	CHPID ID	MOD	TYP	ENTRY SW	PO	CU ID	LA	#	RNG	TYPE	UA	PROC.CSSID	PART. NAME	DEVICE NUM	OS	CHPID ID	MOD	TYP	ENTRY SW	PO	CU ID	LA	#
1	G33XMP.0	TCSS0LP2	8004	Y	20	SHR	FC	FF	06	8004	0007	22	2	FCTC	20	G33XMP.2	TCSS2LP2	8000	Y	21	SHR	FC	FF	07	8000	0006	2
2	G33XMP.0	TCSS0LP2	8006	Y	20	SHR	FC	FF	06	8006	0007	25	2	FCTC	20	G33XMP.2	TCSS2LP5	8000	Y	21	SHR	FC	FF	07	8000	0006	2
3	G33XMP.0	TCSS0LP2	9000	Y	20	SHR	FC	FF	06	9000	0005	4	2	FCTC	00	R31SMP	RAPOS4	9004	Y	10	SHR	FC	FF	05	9004	0006	2
4	G33XMP.0	TCSS0LP2	9002	Y	20	SHR	FC	FF	06	9002	0005	9	2	FCTC	00	R31SMP	RAPMIX9	9004	Y	10	SHR	FC	FF	05	9004	0006	2
5	G33XMP.0	TCSS0LP3	8004	Y	20	SHR	FC	FF	06	8004	0007	22	2	FCTC	20	G33XMP.2	TCSS2LP2	8002	Y	21	SHR	FC	FF	07	8002	0006	3
6	G33XMP.0	TCSS0LP3	8006	Y	20	SHR	FC	FF	06	8006	0007	25	2	FCTC	20	G33XMP.2	TCSS2LP5	8002	Y	21	SHR	FC	FF	07	8002	0006	3
7	G33XMP.0	TCSS0LP3	9000	Y	20	SHR	FC	FF	06	9000	0005	4	2	FCTC	00	R31SMP	RAPOS4	9006	Y	10	SHR	FC	FF	05	9006	0006	3
8	G33XMP.0	TCSS0LP3	9002	Y	20	SHR	FC	FF	06	9002	0005	9	2	FCTC	00	R31SMP	RAPMIX9	9006	Y	10	SHR	FC	FF	05	9006	0006	3
9	G33XMP.2	TCSS2LP2	8000	Y	21	SHR	FC	FF	07	8000	0006	2	2	FCTC	20	G33XMP.0	TCSS0LP2	8004	Y	20	SHR	FC	FF	06	8004	0007	22
10	G33XMP.2	TCSS2LP2	8002	Y	21	SHR	FC	FF	07	8002	0006	3	2	FCTC	20	G33XMP.0	TCSS0LP3	8004	Y	20	SHR	FC	FF	06	8004	0007	22
11	G33XMP.2	TCSS2LP5	8000	Y	21	SHR	FC	FF	07	8000	0006	2	2	FCTC	20	G33XMP.0	TCSS0LP2	8006	Y	20	SHR	FC	FF	06	8006	0007	25
12	G33XMP.2	TCSS2LP5	8002	Y	21	SHR	FC	FF	07	8002	0006	3	2	FCTC	20	G33XMP.0	TCSS0LP3	8006	Y	20	SHR	FC	FF	06	8006	0007	25
13	R31SMP	RAPMIX9	9004	Y	10	SHR	FC	FF	05	9004	0006	2	2	FCTC	00	G33XMP.0	TCSS0LP2	9002	Y	20	SHR	FC	FF	06	9002	0005	9
14	R31SMP	RAPMIX9	9006	Y	10	SHR	FC	FF	05	9006	0006	3	2	FCTC	00	G33XMP.0	TCSS0LP3	9002	Y	20	SHR	FC	FF	06	9002	0005	9
15	R31SMP	RAPOS4	9004	Y	10	SHR	FC	FF	05	9004	0006	2	2	FCTC	00	G33XMP.0	TCSS0LP2	9000	Y	20	SHR	FC	FF	06	9000	0005	4
16	R31SMP	RAPOS4	9006	Y	10	SHR	FC	FF	05	9006	0006	3	2	FCTC	00	G33XMP.0	TCSS0LP3	9000	Y	20	SHR	FC	FF	06	9000	0005	4

Figure 187. CTC Connection Report

Diagnostic messages

The following example shows you messages that might be returned with the report. The diagnostic messages are sorted by severity. For each connection, HCD displays only one messages, even if the connection includes several errors. You first have to correct the first error before the next message is displayed. HCD displays the messages according to the priority as described in "Displaying diagnostic messages" on page 160.

LINE	SEV	MSGID	MESSAGE TEXT
1	E	CBDG750I	Logical address (CUADD) is specified for CU 1010, but CHPID 20 of processor PROC001A is not defined as shared.
3	E	CBDG750I	Logical address (CUADD) is specified for CU 1010, but CHPID 20 of processor PROC001A is not defined as shared.
4	E	CBDG752I	Channel path type error. CHPID 20 of processor PROC001A is connected to a CHPID 11 of processor PROC002 with the same type.
8	E	CBDG752I	Channel path type error. CHPID 11 of processor PROC002 is connected to a CHPID 20 of processor PROC001A with the same type.
11	E	CBDG751I	Device type of device 0805 connected to processor PROC002, CHPID 22 does not match with device type of device 0405 on the other side.
13	E	CBDG750I	Logical address (CUADD) is specified for CU 0108, but CHPID 21 of processor PROC003 is not defined as shared.
18	E	CBDG750I	Logical address (CUADD) is specified for CU 0108, but CHPID 21 of processor PROC003 is not defined as shared.
21	E	CBDG751I	Device type of device 0405 connected to processor PROC003, CHPID 11 does not match with device type of device 0805 on the other side.
6	W	CBDG753I	Wrap around connection detected for processor PROC002 (partition - none -) via CHPID 11 and CHPID 13.
7	W	CBDG753I	Wrap around connection detected for processor PROC002 (partition - none -) via CHPID 13 and CHPID 11.
15	W	CBDG754I	HCD cannot determine connection. No control units and devices match to processor PROC003, partition PART1, CU 1012 and device 1012.
19	W	CBDG754I	HCD cannot determine connection. No control units and devices match to processor PROC003, partition PART2, CU 1012 and device 1012.
5	I	CBDG757I	HCD cannot determine connection. CHPID 10 of processor PROC002 has no dynamic switch defined.
14	I	CBDG756I	HCD cannot determine connection. CHPID 24 of processor PROC002 is connected via chained switches.

Total Messages	Terminating	Error	Warning	Informational
14		8	4	2

Figure 188. Sample of Diagnostic Messages coming with the CTC connection report

I/O Path Report

The I/O Path report shows the physically sensed I/O paths (with physical types) of the active system compared with the logical definitions of the paths (also the object types) of a specific IODF.

- If the sensed I/O path reports a switch, the verification assumes that this is a dynamic switch. It checks whether the defined I/O path in the IODF contains a dynamic switch, and whether the link address corresponds to the output port of the sensed data. If the IODF contains defined switch data, it is verified whether the defined data correspond to the sensed data. If dedicated switches are defined, the I/O Path report always shows differences in the D column.
- When you compare between the sensed and defined data, not all fields are used to determine if there is a difference between the two sides (either an *, or @ is shown in the column D of the report). The data that is used are the CHPID, the

control unit number, the device number, and the switch information of the defined dynamic switches and the sensed dynamic switches.

The I/O path verification checks if the actual system contains the same paths as in the defined I/O configuration. Differences are indicated in the I/O Path report in column D with the following characters:

- 1** The * indicates that differences are found between the sensed and the defined I/O path. Either
 - only sensed data is available, or
 - only defined data in the IODF is available, or
 - the sensed and defined switch data differ.
- 2** The C indicates that the defined and the sensed I/O path are the same, but the defined I/O path is defined to the CSS only.
- 3** An @ is a combination of * and C, and indicates that differences are found between the sensed and the defined I/O path, and that the I/O path is defined to the CSS only.
- 4** The 0 indicates that the defined I/O path in the IODF is defined to the operating system only.

SYSTEM:		SYSPLEX:		I/O PATH REPORT			PROCESSOR: G29	CSS ID: 0	TIME: 14:06	DATE: 2012-10-21	PAGE V- 1		
		Sensed data			IODF data								
--CHP--	DYN.SWITCH	CONTROL UNIT	NUMBER	DEVICE	PATH	-CHP--	-SWITCH-	CHAINED-	DYN SW	--CONTROL UNIT--	DEVICE		
ID	TYPE	PI SW PO	NUMB TYPE	NUMBER	TYPE	ID	TYP PI SW PO	PI SW PO DS LA	LA	L	NUMBER	TYPE	S
00	OSA		0230 9676	0230,2		00	OSA		0230	OSA	0230,2	OSA	
00	OSA		0230 9676	023F	9676-2	00	OSA		0230	OSA	023F	OSAD	
						04	CFS		FFF7	CFS	FF00,2	CFS	@ 3
04	CFS		FFF7 9672-E03	FFE2,2	CFS								*
05	CFS		FFF8 9672-E03	FFE6,2	CFS	05	CFS		FFF8	CFS	FFE6,2	CFS	C 2
						09	BL		0910	3880-3	0910,8	3380	*
						09	BL		0920	3880-3	0920,8	3380	* 1
10	CTC_S	EE AC C3	4010 9672-R72	4010,4	9672-CTC	10	CTC	EE AC C3	AC C3	4010	SCTC	1 4010,4	SCTC
10	CTC_S	EE AC C3	4020 9672-R72	4020,4	9672-CTC	10	CTC	EE AC C3	AC C3	4020	SCTC	2 4020,4	SCTC
10	CTC_S	EE AC CB	4030 9672-R72	4030,4	9672-CTC	10	CTC	EE AC CB	AC CB	4030	SCTC	1 4030,4	SCTC
10	CTC_S	EE AC CB	4040 9672-R72	4040,4	9672-CTC								* 1
11	CNC_S	D0 AE FE	001E 9032-3	001E	9032-3	11	CNC	D0 AE FE	AE FE	001E	9032-3	001E	9032-3
11	CNC_S	D0 AE D8	2000 9394-3	2000,14	9395-B13	11	CNC	D0 AE D8	AE D8	2000	3990-6	2000,14	3390
11	CNC_S	D0 AE D8	2000 9394-3	200E,18		11	CNC	D0 AE D8	AE D8	2000	3990-6	200E,18	3390
12	CNC_?					12	CNC						
13	CNC_?					13	CNC						
14	CNC_S	ED AA D8	0B30 3490-A20	0B30,16	3490-B40	14	CNC	ED AA D8	AA D8	0B30	3490	0B30,16	3490
14	CNC_S	ED AA EA	0B50 3490-C2A	0B50,2	3490-C2A	14	CNC	ED AA EA	AA EA	0B50	3490	0B50,2	3490
						15	CVC		0BA0	3480	0BA0,16	3480	*
15	CNC_S	10 01 FA	2540 2105-E20	2540,37	2105	Y	15	CNC	10 01	01	FA	5 2540,37	3390B
									01	FA	2540	2105	5 2565,27
15	CNC_S	10 01 FB	3A40 2105-F20	3A40,32	2105	Y	15	CNC	10 01	01	FB	5 3A40,32	3390B
									01	FB	3A40	2105	5 3A60,4
15	CNC_S	10 01 FB	3A40 2105-F20	3A60,4		Y	15	CNC	10 01	01	FB	5 3A64,28	3390A
									01	FB	3A40	2105	5 3A60,4
15	CNC_S	10 01 DA	3E80 2105-E20	3E80,13	2105	Y	15	CNC	10 01	01	DA	3E80	2105
									01	DA	3E80	2105	A 3E80,13
									01	DA	3E80	2105	A 3EA0,32
18	CNC_S	E5 AA FE	001A 9032	001A	9032		18	CNC	E5 AA FE	AA FE	001A	9032	001A
18	CNC_S	E5 AA E4	0CC0 3990-6	0CC0,8	3390-A28		18	CNC	E5 AA E4	AA E4	0CC0	3990-6	0CC0,8
18	CNC_S	E5 AA E4	0CC0 3990-6	0CC8,24	3390-B2C		18	CNC	E5 AA E4	AA E4	0CC0	3990-6	0CC8,24
19	CNC_S	F5 AA FE	001A 9032	001A	9032		19	CNC	F5 AA FE	AA FE	001A	9032	001A
1C	CNC_S		0F00	0F00,2			1C	CNC		77	96	0F00	3174
1C	CNC_S		0F02	0F02,2			1C	CNC		77	95	0F02	3174
20	OSA		0210 9676	0214,11			20	OSA		0210	OSA	0214,11	OSA
20	OSA		0210 9676	021F	9676-2		20	OSA		0210	OSA	021F	OSAD
2C	CNC_S	ED AC D8	5100	5100,4			2C	CNC	ED AC D8	AC D8	5100	SCTC	2 5100,4
2F	CNC_S	A0 01 B1	200A 3990-6	0219,2	3390-B3C								*
2F	CNC_S	A0 01 B1	200A 3990-6	021B		Y UNKN							*
40	CNC_S	8F 70 E5	9101 3990-L03	9100,4	3380-AK4		40	CNC	B6 6F 8F 8F 70 E5 70 E5	9101	3990	9100,4	3380
										0180	3172	0180,16	3172
										0640	3990	0640,8	3380
										0740	SCTC	0740,16	SCTC
													0

Figure 189. Example and Legend of an I/O Path Report (Part 1 od 2)

KEY	KEY DESCRIPTION
SENSED DATA	- I/O CONFIGURATION DATA RETRIEVED BY SENSING I/O
CHP ID	- SENSED CHANNEL PATH ID
CHP TYPE	- SENSED CHANNEL PATH TYPE
DYN.SWITCH PI	- SENSED DYN. SWITCH INPUT PORT (CHANNEL PATH SIDE)
DYN.SWITCH SW	- SENSED DYNAMIC SWITCH ID
DYN.SWITCH PO	- SENSED DYN. SWITCH OUTPUT PORT (CONTROL UNIT SIDE)
CONTROL UNIT NUMB-	SENSED CONTROL UNIT NUMBER
CONTROL UNIT TYPE-	SENSED CONTROL UNIT TYPE
DEVICE NUMBER	- SENSED DEVICE NUMBER AND RANGE
DEVICE TYPE	- SENSED DEVICE TYPE
DEVICE O	- SENSED DEVICE(S) OFFLINE INDICATOR
	Y - OFFLINE
	BLANK - NOT OFFLINE
PATH STAT	- IF BLANK, SENSED I/O PATH IS ONLINE
	IF OFFL, SENSED I/O PATH IS OFFLINE
	IF UNKN, STATUS OF SENSED I/O PATH IS UNKNOWN
	IF ONLINE, SINGLE POINT OF FAILURE INDICATION
	POS 1: BLANK, U, OR NUMBER BETWEEN 3 AND 8
	3: BOOK, 4: CAGE, 5: FAILOVER DOMAIN
	6: FANOUT, 7: DOMAIN, 8: SECONDARY STI/STI
	U: NOT EXECUTED
	POS 2: C, IF CU I/F SHARE SINGLE POINT OF FAILURE
	N, IF CU I/F NOT CHECKED
	POS 3: P, IF DEVICE HAS ONLY ONE PATH ONLINE
	POS 4: S, IF ALL PATHES GO THROUGH ONE SWITCH
IODF DATA	- DATA RETRIEVED FROM IODF
CHP ID	- DEFINED CHANNEL PATH ID
CHP TYPE	- DEFINED CHANNEL PATH TYPE
SWITCH PI	- DEFINED SWITCH INPUT PORT (CHANNEL PATH SIDE)
SWITCH SW	- DEFINED SWITCH ID
SWITCH PO	- DEFINED SWITCH OUTPUT PORT (CONTROL UNIT SIDE)
CHAINED PI	- DEFINED CHAINED SWITCH INPUT PORT
CHAINED SW	- DEFINED CHAINED SWITCH ID
CHAINED PO	- DEFINED CHAINED SWITCH OUTPUT PORT
DYNSW SW	- DEFINED DYNAMIC SWITCH FOR CHANNEL PATH
DYNSW LA	- DEFINED DYNAMIC LINK ADDRESS FOR CONTROL UNIT
CONTROL UNIT NUMB-	DEFINED CONTROL UNIT NUMBER
CONTROL UNIT TYPE-	DEFINED CONTROL UNIT TYPE
CONTROL UNIT L	- DEFINED CONTROL UNIT LOGICAL ADDRESS
DEVICE NUMBER	- DEFINED DEVICE NUMBER AND RANGE
DEVICE TYPE	- DEFINED DEVICE TYPE
DEVICE S	- DEFINED DEVICE SUBCHANNEL SET ID
D	- DIFFERENCES FOUND
	IF C, ONLY DEFINED TO CHANNEL SUBSYSTEM
	IF O, ONLY DEFINED TO OS CONFIGURATION
	IF *, DIFFERENCES WERE FOUND BETWEEN THE 2 SIDES
	IF @, INDICATES C AND * FROM ABOVE

Figure 190. Example and Legend of an I/O Path Report (Part 2 of 2)

DYN.SWITCH

Contains either the switch information for the dynamic switch in the path or the dedicated switch if there is no dynamic switch.

PATH STAT

Represents the status of the I/O path between the CHPID and the devices that it is connected to.

blank Represents the situations where the sensed I/O path is online. For more information, refer to “The I/O path list” on page 229.

OFFL Represents the situations where there is no I/O path to be found or when the I/O path is offline.

UNKN Represents the situations where the I/O path is currently in a pending state or when z/OS is running as a guest on a z/VM system and the path status can not be obtained from the system.

Supported hardware report

The Supported Hardware Report shows the actual status of the hardware supported in your installation. It shows the following supported hardware:

- Processors
- Control units
- Devices including:
 - Device characteristics
 - Control unit attachments
- MVS devices including:
 - Device capabilities
 - Parameters (with selection values) / features
- VM devices including
 - device capabilities
 - parameters/features

The following figure shows you an example of a supported hardware report. The example may differ from the report you get on your system, because the data depend on the installed processor support modules and UIMs.

TYPE-MODEL		MOD	SUPLEVID	IOCP	SYSTEM	SUPPORTED HARDWARE - PROCESSORS SUPPORTED CHPID TYPES	WI	RI	DP	FCT	CHPID	CU	LCU	SUBCH	LPAR	CSS	SCHS
2064-1C1	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C1	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C2	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C2	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C3	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C3	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C4	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C4	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C5	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C5	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C6	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C6	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C7	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C7	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C8	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C8	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C9	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-1C9	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-100	201	H000931	IYP	2064,2	CFR,CFS,CBR,CBS,CFP,CBP,ICP	Y	Y	N	N		64	8192	4096	64512	15		
2064-101	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-101	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-102	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-102	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-103	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-103	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N		256	8192	4096	64512	15		
2064-104	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,	Y	Y	N	N		256	8192	4096	64512	15		

Figure 191. Supported Hardware Report (Part 1 of 30)

					SUPPORTED HARDWARE - PROCESSORS	TIME: 14:21 DATE: 2016-05-11 PAGE X- 2										
TYPE-MODEL	MOD	SUPLEVID	IOCP	SYSTEM	SUPPORTED CHPID TYPES	WI	RI	DP	FCT	CHPID	CU	LCU	SUBCH	LPAR	CSS	SCHS
					FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP											
2064-104	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-105	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-105	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-106	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-106	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-107	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-107	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-108	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-108	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-109	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-109	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-110	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-110	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-111	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-111	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-112	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-112	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-113	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-113	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-114	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-114	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-115	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-115	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-116	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA, FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP, ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		

Figure 192. Supported Hardware Report (Part 2 of 30)

					SUPPORTED HARDWARE - PROCESSORS	TIME: 14:21 DATE: 2016-05-11 PAGE X- 3										
TYPE-MODEL	MOD	SUPLEVID	IOCP	SYSTEM	SUPPORTED CHPID TYPES	WI	RI	DP	FCT	CHPID	CU	LCU	SUBCH	LPAR	CSS	SCHS
2064-116	202	H010931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C1	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C1	202	H020415	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C2	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C2	202	H020415	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C3	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C3	202	H020415	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C4	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C4	202	H020415	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C5	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C5	202	H020415	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C6	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C6	202	H020415	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C7	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C7	202	H020415	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C8	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C8	202	H020415	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C9	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-2C9	202	H020415	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-210	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-210	202	H020415	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-211	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-211	202	H020415	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-212	200	H000931	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-212	202	H020415	IYP	2064,1	BL,BY,CNC,CTC,CVC,CFR,CFS,CBY,OSA,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	256	8192	4096	64512	15		

Figure 193. Supported Hardware Report (Part 3 of 30)

					SUPPORTED HARDWARE - PROCESSORS	TIME: 14:21 DATE: 2016-05-11 PAGE X- 4										
TYPE-MODEL	MOD	SUPLEVID	IOCP	SYSTEM	SUPPORTED CHPID TYPES	WI	RI	DP	FCT	CHPID	CU	LCU	SUBCH	LPAR	CSS	SCHS
2064-213	200	H000931	IYP	2064,1	ICP, IQD, FCP BL, BY, CNC, CTC, CVC, CFR, CFS, CBY, OSA, FC, FCV, CBR, CBS, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-213	202	H020415	IYP	2064,1	BL, BY, CNC, CTC, CVC, CFR, CFS, CBY, OSA, FC, FCV, CBR, CBS, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-214	200	H000931	IYP	2064,1	BL, BY, CNC, CTC, CVC, CFR, CFS, CBY, OSA, FC, FCV, CBR, CBS, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-214	202	H020415	IYP	2064,1	BL, BY, CNC, CTC, CVC, CFR, CFS, CBY, OSA, FC, FCV, CBR, CBS, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-215	200	H000931	IYP	2064,1	BL, BY, CNC, CTC, CVC, CFR, CFS, CBY, OSA, FC, FCV, CBR, CBS, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-215	202	H020415	IYP	2064,1	BL, BY, CNC, CTC, CVC, CFR, CFS, CBY, OSA, FC, FCV, CBR, CBS, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-216	200	H000931	IYP	2064,1	BL, BY, CNC, CTC, CVC, CFR, CFS, CBY, OSA, FC, FCV, CBR, CBS, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2064-216	202	H020415	IYP	2064,1	BL, BY, CNC, CTC, CVC, CFR, CFS, CBY, OSA, FC, FCV, CBR, CBS, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0A1	206	H020331	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0A1	208	H020731	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0A2	206	H020331	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0A2	208	H020731	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0B1	206	H020331	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0B1	208	H020731	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0CF	207	H020331	IYP	2066,2	CFR, CFS, CFP, CBP, ICP	Y	Y	N	N	64	8192	4096	64512	15		
2066-0C1	206	H020331	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0C1	208	H020731	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0E1	206	H020331	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0E1	208	H020731	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0LF	206	H020331	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0LF	208	H020731	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0X2	206	H020331	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-0X2	208	H020731	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-001	206	H020331	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-001	208	H020731	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-002	206	H020331	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-002	208	H020731	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-003	206	H020331	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-003	208	H020731	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-004	206	H020331	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2066-004	208	H020731	IYP	2066,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	256	8192	4096	64512	15		
2084-A08	210	H030530	ICP	2084,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV, CBR, CBS, OSD, OSE, CFP, CBP, ICP, IQD, FCP	Y	Y	N	N	512	8192	4096	64512	30	2	
2084-A08	211	H040331	ICP	2084,1	CNC, CTC, CVC, CFR, CFS, CBY, FC, FCV,	Y	Y	Y	N	1024	8192	4096	64512	30	4	

Figure 194. Supported Hardware Report (Part 4 of 30)

TYPE-MODEL		MOD	SUPLEVID	IOCP	SYSTEM	SUPPORTED HARDWARE - PROCESSORS SUPPORTED CHPID TYPES	WI	RI	DP	FCT	CHPID	CU	LCU	SUBCH	LPAR	CSS	SCHS
2084-B16	210	H030530	ICP	2084,1	CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC	Y	Y	N	N	512	8192	4096	64512	30	2		
2084-B16	211	H040331	ICP	2084,1	CNC,CTC,CVC,CFR,CFS,CBY,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	Y	N	1024	8192	4096	64512	30	4		
2084-C24	210	H030530	ICP	2084,1	CNC,CTC,CVC,CFR,CFS,CBY,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	512	8192	4096	64512	30	2		
2084-C24	211	H040331	ICP	2084,1	CNC,CTC,CVC,CFR,CFS,CBY,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC	Y	Y	Y	N	1024	8192	4096	64512	30	4		
2084-D32	210	H030530	ICP	2084,1	CNC,CTC,CVC,CFR,CFS,CBY,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP	Y	Y	N	N	512	8192	4096	64512	30	2		
2084-D32	211	H040331	ICP	2084,1	CNC,CTC,CVC,CFR,CFS,CBY,FC,FCV,CBR,CBS,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC	Y	Y	Y	N	1024	8192	4096	64512	30	4		
2086-A04	212	H040331	ICP	2086,1	CNC,CTC,CVC,CFR,CFS,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC	Y	Y	Y	N	512	8192	4096	64512	30	2		
2094-S08	213	H050331	ICP	2094,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2094-S08	216	H070331	ICP	2094,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2094-S18	213	H050331	ICP	2094,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2094-S18	216	H070331	ICP	2094,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2094-S28	213	H050331	ICP	2094,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2094-S28	216	H070331	ICP	2094,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2094-S38	213	H050331	ICP	2094,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2094-S38	216	H070331	ICP	2094,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2094-S54	213	H050331	ICP	2094,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2094-S54	216	H070331	ICP	2094,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2096-R07	214	H060930	ICP	2096,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN	Y	Y	Y	N	512	8192	4096	65280	15	2	2	
2096-R07	218	H070331	ICP	2096,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	512	8192	4096	65280	15	2	2	
2096-S07	215	H060930	ICP	2096,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN	Y	Y	Y	N	512	8192	4096	65280	30	2	2	
2096-S07	219	H070331	ICP	2096,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	512	8192	4096	65280	30	2	2	
2097-E12	217	H080130	ICP	2097,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2097-E12	241	H080131	ICP	2097,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2097-E26	217	H080130	ICP	2097,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2097-E26	241	H080131	ICP	2097,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2097-E40	217	H080130	ICP	2097,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2097-E40	241	H080131	ICP	2097,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2097-E56	217	H080130	ICP	2097,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2097-E56	241	H080131	ICP	2097,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2097-E64	217	H080130	ICP	2097,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	1024	8192	4096	65280	60	4	2	
2098-E10	240	H081130	ICP	2098,1	CNC,CTC,CVC,CBY,FC,FCV,OSD,OSE,CFP,CBP,ICP,IQD,FCP,OSC,OSN,CIB	Y	Y	Y	N	512	8192	4096	65280	30	2	2	
2817-M15	242	H100331	ICP	2817,1	CNC,CTC,CVC,CBY,FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	Y	Y	Y	N	1024	8192	4096	65280	60	4	3	
2817-M32	242	H100331	ICP	2817,1	CNC,CTC,CVC,CBY,FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	Y	Y	Y	N	1024	8192	4096	65280	60	4	3	

Figure 195. Supported Hardware Report (Part 5 of 30)

				SUPPORTED HARDWARE - PROCESSORS		TIME: 14:21 DATE: 2016-05-11 PAGE X- 6																	
TYPE-MODEL	MOD	SUPLEVID	IOCP	SYSTEM	SUPPORTED	CHPID	CU	LCU	SUBCH	LPAR	CSS	SCHS	WI	RI	DP	FCT	CHPID	CU	LCU	SUBCH	LPAR	CSS	SCHS
2817-M49	242	H100331	ICP	2817,1	CNC,CTC,CVC,CBY,FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	N	1024	8192	4096	65280	60	4	3
2817-M66	242	H100331	ICP	2817,1	CNC,CTC,CVC,CBY,FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	N	1024	8192	4096	65280	60	4	3
2817-M80	242	H100331	ICP	2817,1	CNC,CTC,CVC,CBY,FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	N	1024	8192	4096	65280	60	4	3
2818-M05	245	H110331	ICP	2818,1	CNC,CTC,CVC,CBY,FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	512	8192	4096	65280	30	2	2	Y	Y	Y	N	512	8192	4096	65280	30	2	2
2818-M10	245	H110331	ICP	2818,1	CNC,CTC,CVC,CBY,FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	512	8192	4096	65280	30	2	2	Y	Y	Y	N	512	8192	4096	65280	30	2	2
2827-HA1	244	H120331	ICP	2827,1	FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	N	1024	8192	4096	65280	60	4	3
2827-HA1	247	H130331	ICP	2827,1	FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	Y	1024	8192	4096	65280	60	4	3
2827-H20	244	H120331	ICP	2827,1	FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	N	1024	8192	4096	65280	60	4	3
2827-H20	247	H130331	ICP	2827,1	FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	Y	1024	8192	4096	65280	60	4	3
2827-H43	244	H120331	ICP	2827,1	FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	N	1024	8192	4096	65280	60	4	3
2827-H43	247	H130331	ICP	2827,1	FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	Y	1024	8192	4096	65280	60	4	3
2827-H66	244	H120331	ICP	2827,1	FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	N	1024	8192	4096	65280	60	4	3
2827-H66	247	H130331	ICP	2827,1	FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	Y	1024	8192	4096	65280	60	4	3
2827-H89	244	H120331	ICP	2827,1	FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	N	1024	8192	4096	65280	60	4	3
2827-H89	247	H130331	ICP	2827,1	FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	1024	8192	4096	65280	60	4	3	Y	Y	Y	Y	1024	8192	4096	65280	60	4	3
2828-H06	248	H130331	ICP	2828,1	FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	512	8192	4096	65280	30	2	2	Y	Y	Y	Y	512	8192	4096	65280	30	2	2
2828-H13	248	H130331	ICP	2828,1	FC,OSD,OSE,CFP,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM	512	8192	4096	65280	30	2	2	Y	Y	Y	Y	512	8192	4096	65280	30	2	2
2964-LC9	103	H160310	ICP	2964,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-LC9	104	H160930	ICP	2964,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-LE1	103	H160310	ICP	2964,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-LE1	104	H160930	ICP	2964,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-L30	103	H160310	ICP	2964,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-L30	104	H160930	ICP	2964,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-L63	103	H160310	ICP	2964,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-L63	104	H160930	ICP	2964,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-L96	103	H160310	ICP	2964,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-L96	104	H160930	ICP	2964,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-NC9	102	H160930	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM,CS5	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-NC9	250	H150111	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM,CS5	1024	8192	4096	65280	90	6	4	Y	Y	Y	Y	1024	8192	4096	65280	90	6	4
2964-NC9	252	H160310	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM,CS5	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-NE1	102	H160930	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM,CS5	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-NE1	250	H150111	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM,CS5	1024	8192	4096	65280	90	6	4	Y	Y	Y	Y	1024	8192	4096	65280	90	6	4
2964-NE1	252	H160310	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM,CS5	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-N30	102	H160930	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM,CS5	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-N30	250	H150111	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM,CS5	1024	8192	4096	65280	90	6	4	Y	Y	Y	Y	1024	8192	4096	65280	90	6	4
2964-N30	252	H160310	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM,CS5	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-N63	102	H160930	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM,CS5	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-N63	250	H150111	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM,CS5	1024	8192	4096	65280	90	6	4	Y	Y	Y	Y	1024	8192	4096	65280	90	6	4
2964-N63	252	H160310	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN,CIB,OSX,OSM,CS5	1536	8192	4096	65280	90	6	4	Y	Y	Y	Y	1536	8192	4096	65280	90	6	4

Figure 196. Supported Hardware Report (Part 6 of 30)

					SUPPORTED HARDWARE - PROCESSORS		TIME: 14:21 DATE: 2016-05-11 PAGE X- 7										
TYPE-MODEL	MOD	SUPLEVID	IOCP	SYSTEM	SUPPORTED	CHPID TYPES	WI	RI	DP	FCT	CHPID	CU	LCU	SUBCH	LPAR	CSS	SCHS
2964-N96	102	H160930	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN, CIB,OSX,OSM,CS5		Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2964-N96	250	H150111	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN, CIB,OSX,OSM,CS5		Y	Y	Y	Y	1024	8192	4096	65280	90	6	4
2964-N96	252	H160310	ICP	2964,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN, CIB,OSX,OSM,CS5		Y	Y	Y	Y	1536	8192	4096	65280	90	6	4
2965-L10	108	H160310	ICP	2965,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM		Y	Y	Y	Y	768	8192	4096	65280	45	3	3
2965-L10	111	H160930	ICP	2965,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM		Y	Y	Y	Y	768	8192	4096	65280	45	3	3
2965-L20	108	H160310	ICP	2965,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM		Y	Y	Y	Y	768	8192	4096	65280	45	3	3
2965-L20	111	H160930	ICP	2965,1	FC,OSD,OSE,IQD,FCP,OSC,OSN,OSX,OSM		Y	Y	Y	Y	768	8192	4096	65280	45	3	3
2965-N10	109	H160930	ICP	2965,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN, CIB,OSX,OSM,CS5		Y	Y	Y	Y	768	8192	4096	65280	45	3	3
2965-N10	253	H160310	ICP	2965,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN, CIB,OSX,OSM,CS5		Y	Y	Y	Y	768	8192	4096	65280	45	3	3
2965-N20	109	H160930	ICP	2965,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN, CIB,OSX,OSM,CS5		Y	Y	Y	Y	768	8192	4096	65280	45	3	3
2965-N20	253	H160310	ICP	2965,1	FC,OSD,OSE,ICP,IQD,FCP,OSC,OSN, CIB,OSX,OSM,CS5		Y	Y	Y	Y	768	8192	4096	65280	45	3	3

Figure 197. Supported Hardware Report (Part 7 of 30)

		SUPPORTED HARDWARE - PROCESSORS		TIME: 14:21 DATE: 2016-05-11 PAGE X- 8			
KEY		KEY	DESCRIPTION				
---		-----					
TYPE-MODEL	-	SUPPORTED PROCESSOR TYPE					
MOD	-	INDEX OF PROCESSOR SUPPORT MODULE					
SUPLEVID	-	SUPPORT LEVEL ID					
IOCP	-	PREFIX OF IOCP PROGRAM FOR THIS SUPPORT LEVEL					
SYSTEM	-	IOCP SYSTEM VALUE					
SUPPORTED CHPID TYPES	-	LIST OF CHANNEL PATH TYPES WHICH ARE SUPPORTED BY THIS CONTROL UNIT					
WI	-	WRITE IOCDs REGARDLESS OF CPC TYPE					
RI	-	RECEIVE IOCDs WRITTEN REGARDLESS OF CPC TYPE					
DP	-	SUPPORT OF DYNAMIC PARTITIONS					
FCT	-	SUPPORT OF PCIE FUNCTIONS					
CHPID	-	MAXIMUM NUMBER OF SUPPORTED CHPIDS					
CU	-	MAXIMUM NUMBER OF ATTACHABLE CONTROL UNITS					
LCU	-	MAXIMUM NUMBER OF LOGICAL CONTROL UNITS (PER CHANNEL SUBSYSTEM)					
SUBCH	-	MAXIMUM NUMBER OF SUBCHANNELS (PER CHANNEL SUBSYSTEM)					
LPAR	-	MAXIMUM NUMBER OF LOGICAL PARTITIONS					
CSS	-	MAXIMUM NUMBER OF CHANNEL SUBSYSTEMS (XMP PROCESSOR ONLY)					
SCHS	-	MAXIMUM NUMBER OF SUBCHANNEL SETS					
Y	-	PROCESSOR HAS THE CAPABILITY					
N	-	CAPABILITY IS NOT AVAILABLE					

Figure 198. Supported Hardware Report (Part 8 of 30)

TIME: 14:21 DATE: 2016-05-11 PAGE X- 9

TYPE-MODEL	UIM U	PROTCL	DP	IO	SUPPORTED CHPID TYPES	SUPPORTED HARDWARE - CONTROL UNITS (1) ATTACHABLE DEVICES
AFP1	022 N	D,S,S4	S	2	BL, BY, CNC, CVC, CBY, FC, FCV, EIO	AFP1, AFP1-0, 2710, 3160, 3170, 3300, 3825, 3827, 3828, 3829, 3831, 3835, 3900, 3935, 4000, 4100, 4370
CTC	014 N	D	D	2	BL, CVC, EIO	CTC, CTCA
DUMMY	050 N	D,S,S4	D	2	BL, BY, CNC, CTC, CVC, CBY, FCV, EIO	DUMMY
FCP	254 N				FCP	FCP
FCTC	014 N			2	FC	FCTC
IQD	058 N				IQD	IQD
NOCHECK	256 N	D,S,S4	S	2	BL, BY, CNC, CTC, CVC, IOC, CBY, FC, FCV, EIO	AFP1-0, BCTC, BSC1, BSC2, BSC3, CTC, CTCA, DUMMY, FBA, FB512, FCTC, HFGD, ICABSCA, ICAELAN, ICAHDL, ICAILAN, ICASDL, ICATELE2, RS6K, SCTC, SWCH, TWX, WTTA, 1030, 1050, 1050X, 115A, 1287, 1288, 1403-N1, 1403-2, 1403-3, 1403-5, 1403-7, 2032, 2250-3, 2305-2, 2501-B1, 2501-B2, 2540P-1, 2540R-1, 2701, 2710, 2740, 2740C, 2741C, 2741P, 3088, 3151, 3160, 3161, 3162, 3163, 3167, 3170, 3172, 3174, 3178, 3179, 3180, 3180-1, 3190, 3191, 3192, 3192-F, 3193, 3194, 3203-4, 3203-5, 3211, 3215, 3250, 3251, 3262-13, 3262-3, 3262-5, 3268-2, 3270, 3270-X, 3272, 3274, 3277-1, 3277-2, 3278-1, 3278-2, 3278-2A, 3278-3, 3278-4, 3278-5, 3279-S2B, 3279-S3G, 3279-2, 3279-2A, 3279-2B, 3279-2C, 3279-2X, 3279-3, 3279-3A, 3279-3B, 3279-3X, 3284-1, 3284-2, 3286-1, 3286-2, 3287-1, 3287-1C, 3287-2, 3287-2C, 3288, 3289-1, 3289-2, 3290, 3300, 3330-1, 3330-11, 3330-2, 3330V, 3333-1, 3333-11, 3340, 3344, 3350, 3370, 3375, 3380, 3380-CJ2, 3380A, 3380B, 3390, 3390A, 3390B, 3390C, 3390S, 3420-3, 3420-4, 3420-5, 3420-6, 3420-7, 3420-8, 3422, 3423, 3424, 3430, 3471, 3472, 3480, 3481, 3482, 3483, 3490, 3505, 3525, 3540, 3590, 3704, 3705, 3720, 3725, 3737, 3745, 3746, 3767-1, 3767-2, 3791L, 3800-1, 3800-3, 3800-6, 3800-8, 3812, 3816, 3825, 3827, 3828, 3829, 3831, 3835, 3851, 3886, 3890, 3895, 3900, 3935, 3995, 3995-SDA, 3995-151, 3995-153, 4000, 4100, 4224, 4245, 4248, 4250, 4370, 5080, 5081, 5210, 6090, 6091, 6262, 7171, 7770-3, 8232, 83B3, 9032, 9032-3, 9032-5, 9033, 9332-40, 9332-42, 9332-60, 9332-62, 9335-B1, 9336-10, 9336-20, 9345, 9348-1
OSA	058 N				OSA, OSD, OSE	CONTROL, OSA, OSAD
OSC	058 N				OSC	3215, 3270-X, 3286, 3287, 3287-1, 3287-1C, 3287-2, 3287-2C
OSM	058 N				OSM	OSA-M, OSAD
OSN	058 N				OSN	OSAD, OSN, 3745
OSX	058 N				OSX	OSA-X, OSAD
RS6K	056 N	D,S,S4	S4	2	BL, CNC, CVC, FCV, EIO	RS6K
RS6K-2	056 N			2	CNC, FCV	RS6K
SCTC	014 N			2	CNC, CTC, FCV	BCTC, CTCA, SCTC
SWCH	051 Y			2	CNC, FCV	SWCH
1287	032 N	D	D	2	BL, BY, CVC, CBY, EIO	1287
1288	032 N	D	D	2	BL, BY, CVC, CBY, EIO	1288
1750	002 Y	S,S4	S	2	FC	3380, 3380A, 3380B, 3390, 3390A, 3390B, 3390D, 3390S
2032	051 Y			2	FC	SWCH, 2032
2105	002 Y	S,S4	S	2	CNC, FC, FCV	3380, 3380A, 3380B, 3390, 3390A, 3390B, 3390D, 3390S
2107	002 Y	S,S4	S	2	CNC, FC, FCV	3380, 3380A, 3380B, 3390, 3390A, 3390B, 3390D, 3390S
2107-FBA	002 Y	S,S4	S	2	CNC, FC, FCV	FBA
2501	012 N	D	D	2	BL, BY, CBY, EIO	2501
2701	024 N	D	D	2	BL, BY, CVC, CBY, EIO	BSC1, BSC2, BSC3, TWX, WTTA, 1030, 1050, 1050X, 115A, 2701, 2740, 2740C, 2740X, 2741, 2741C, 2741P, 3151, 3161, 3162, 3163, 3167, 3767-1, 3767-2, 83B3
2710	022 N	D,S,S4	S	2	BL, CNC, CVC, FC, FCV, EIO	AFP1-0, 2710
2821	012 N	D	D	2	BL, BY, CVC, CBY, EIO	1403, 2540, 2540P-1, 2540R-1
2835-2	269 N	D	D	2	BL, CVC, EIO	2305-2
2840-2	277 N	D	D	2	BL, BY, CBY, EIO	2250-3
3088	014 N	D,S,S4	S	2	BL, CVC, EIO	CTC, CTCA, 3088
3160	022 N	D,S,S4	S	2	BL, BY, CNC, CVC, CBY, FCV, EIO	AFP1-0, 3160
3170	022 N	D,S,S4	S	2	BL, BY, CNC, CVC, CBY, FC, FCV, EIO	AFP1-0, 3170
3172	057 N	D,S,S4	S4	2	BL, CNC, CVC, FCV, EIO	3172
3174	027 N	D,S	D	2	BL, BY, CNC, CVC, CBY, FCV, EIO	3151, 3174, 3178, 3179, 3180, 3190, 3191, 3192, 3193, 3194, 3262, 3262-13, 3262-3, 3268, 3270, 3270-X, 3278, 3279, 3286, 3287, 3289, 3290, 3471, 3472, 3481, 3482, 3483, 3791L, 3812, 3816, 4201, 4202, 4207, 4208, 4224, 4234, 4245-D12, 4245-D20, 4250, 5210
3203	012 N	D	D	2	BL, BY, CVC, CBY, EIO	3203
3258	277 N	D	D	1	BL, CVC, EIO	2250-3, 3250, 3251
3262	268 N	D	D	2	BL, BY, CVC, CBY, EIO	3262-5
3272	027 N	D	D	2	BL, BY, CVC, CBY, EIO	3151, 3178, 3179, 3180, 3190, 3191, 3192, 3193, 3194, 3262, 3262-13, 3262-3, 3268, 3270, 3270-X, 3272, 3274, 3277, 3278, 3279, 3284, 3284-1, 3284-2, 3286, 3287, 3288, 3289, 3290, 3791L, 3812, 3816, 4224, 4234, 4245-D12, 4245-D20, 4250, 5210
3274	027 N	D	D	2	BL, BY, CVC, CBY, EIO	3151, 3178, 3179, 3180, 3190, 3191, 3192, 3193, 3194, 3262, 3262-13, 3262-3, 3268, 3270, 3270-X, 3274, 3277, 3278, 3279, 3284, 3284-1, 3284-2, 3286, 3287, 3288, 3289, 3290, 3471, 3472, 3481, 3482, 3483, 3791L, 3812, 3816, 4224, 4234, 4245-D12, 4245-D20, 4250, 5210
3300	022 N	D,S,S4	S	2	BL, BY, CNC, CVC, CBY, FC, FCV, EIO	AFP1-0, 3300
3380-CJ2	002 N	S	S	2	BL, CVC, EIO	3380, 3380-CJ2

Figure 199. Supported Hardware Report (Part 9 of 30)

		SUPPORTED HARDWARE - CONTROL UNITS (1)				TIME: 14:21 DATE: 2016-05-11 PAGE X- 10
TYPE-MODEL	UIM U PROTCL DP IO	SUPPORTED CHPID TYPES	ATTACHABLE DEVICES			
3422	005 N D,S	D 1	BL,CVC,EIO	3422		
3423	005 N D,S,S4	D 1	BL,CVC,EIO	3423		
3424	005 N S	S 2	BL,CVC,EIO	3424		
3430	005 N D	D 1	BL,CVC,EIO	3430		
3480	005 Y D,S,S4	S 2	BL,CNC,CVC,FCV,EIO	3480		
3490	005 Y D,S,S4	S 2	BL,CNC,CVC,ISD,FC,FCV,EIO	3490		
3490-C1A	005 Y D,S,S4	S 2	BL,CNC,CVC,FC,FCV,EIO	3490		
3490-C10	005 Y D,S,S4	S 2	BL,CNC,CVC,ISD,FC,FCV,EIO	3490		
3490-C11	005 Y D,S,S4	S 2	BL,CNC,CVC,ISD,FC,FCV,EIO	3490		
3490-C2A	005 Y D,S,S4	S 2	BL,CNC,CVC,FC,FCV,EIO	3490		
3490-C22	005 Y D,S,S4	S 2	BL,CNC,CVC,FC,FCV,EIO	3490		
3505	012 N D	D 2	BL,BY,CVC,CBY,EIO	3505,3525		
3540	032 N D	D 2	BL,BY,CVC,CBY,EIO	3540		
3590	005 Y D,S,S4	S 2	BL,CNC,CVC,FC,FCV,EIO	3590		
3704	023 N D	D 2	BY,CBY	BSC1,BSC2,BSC3,TWX,WTTA,1030,1050,1050X,115A,2740,2740C,2740X,2741C,2741P,3151,3161,3162,3163,3167,3174,3232,3271,3274,3276,3704,3767-1,3767-2,3791L,83B3		
3705	023 N D	D 2	BL,BY,CNC,CVC,CBY,FCV,EIO	BSC1,BSC2,BSC3,NCP,TWX,WTTA,1030,1050,1050X,115A,2740,2740C,2740X,2741C,2741P,3151,3161,3162,3163,3167,3174,3232,3271,3274,3276,3705,3767-1,3767-2,3791L,83B3		
3720	023 N D	D 2	BL,BY,CVC,CBY,EIO	BSC1,BSC2,BSC3,NCP,TWX,WTTA,1030,1050,1050X,115A,2740,2740C,2740X,2741C,2741P,3151,3161,3162,3163,3167,3174,3232,3271,3274,3276,3720,3767-1,3767-2,3791L,83B3		
3725	023 N D	D 2	BL,BY,CVC,CBY,EIO	BSC1,BSC2,BSC3,NCP,TWX,WTTA,1030,1050,1050X,115A,2740,2740C,2740X,2741C,2741P,3151,3161,3162,3163,3167,3174,3232,3271,3274,3276,3725,3767-1,3767-2,3791L,83B3		
3737	014 N D,S	S 2	BL,CVC,EIO	CTC,CTCA,3737,4753		
3745	023 N D,S	D 2	BL,BY,CNC,CVC,CBY,FCV,EIO	BSC1,BSC2,BSC3,NCP,TWX,WTTA,1030,1050,1050X,115A,2740,2740C,2740X,2741C,2741P,3151,3161,3162,3163,3167,3174,3232,3271,3274,3276,3745,3767-1,3767-2,3791L,83B3		
3746	023 N D,S	D 2	BL,BY,CNC,CVC,CBY,FCV,EIO	BSC1,BSC2,BSC3,NCP,TWX,WTTA,1030,1050,1050X,115A,2740,2740C,2740X,2741C,2741P,3174,3232,3271,3274,3276,3745,3767-1,3767-2,3791L,83B3		
3791L	027 N D,S	D 2	BL,BY,CNC,CVC,CBY,FCV,EIO	3151,3178,3179,3180,3190,3191,3192,3193,3194,3262,3262-13,3262-3,3268,3270,3270-X,3272,3277,3278,3279,3284-1,3284-2,3286,3287,3288,3289,3290,3791L,3812,3816,4224,4234,4245-D12,4245-D20,4250,5210		
3800-1	011 N D,S	D 2	BL,BY,CVC,CBY,EIO	3800-1		
3800-3	011 N D,S	D 2	BL,BY,CVC,CBY,EIO	AFP1,3800-3		
3800-6	011 N D,S	D 2	BL,BY,CVC,CBY,EIO	AFP1,3800-6		
3800-8	011 N D,S	D 2	BL,BY,CVC,CBY,EIO	AFP1,3800-8		
3803	005 N D	D 1	BL,CVC,EIO	3420		
3811	012 N D	D 2	BL,BY,CVC,CBY,EIO	3211,3216		
3820	022 N D,S4	D 2	BL,BY,CNC,CVC,CBY,FC,FCV,EIO	AFP1,3820		
3825	022 N D,S,S4	S 2	BL,BY,CNC,CVC,CBY,FC,FCV,EIO	AFP1,AFP1-0,3825		
3827	022 N D,S	S 2	BL,BY,CVC,CBY,EIO	AFP1,AFP1-0,3827		
3828	022 N D,S	S 2	BL,BY,CVC,CBY,EIO	AFP1,AFP1-0,3828		
3829	022 N D,S,S4	S 2	BL,BY,CNC,CVC,CBY,FCV,EIO	AFP1-0,3829		
3830-2	257 N D	D 2	BL,CVC,EIO	3330-1,3330-11,3330-2,3333-1,3333-11,3340,3344,3350		
3830-3	271 N D	D 2	BL,CVC,EIO	3330-11,3330V,3350		
3831	022 N D,S	S 2	BL,BY,CVC,CBY,EIO	AFP1-0,3831		
3835	022 N D,S	S 2	BL,BY,CVC,CBY,EIO	AFP1,AFP1-0,3835		
3851	271 N D	D 2	BL,BY,CVC,CBY,EIO	3851		
3880-1	258 N D,S	S 2	BL,CVC,EIO	3330-1,3330-11,3330-2,3333-1,3333-11,3340,3344,3350,3370,3375		
3880-11	259 N D,S	S 2	BL,CVC,EIO	3370,3375		
3880-13	002 N D,S	S 2	BL,CVC,EIO	3380		
3880-2	002 N D,S	S 2	BL,CVC,EIO	3330-1,3330-11,3330-2,3333-1,3333-11,3340,3350,3370,3375,3380		
3880-23	002 N D,S,S4	S 2	BL,CVC,EIO	3380		
3880-3	002 N D,S	S 2	BL,CVC,EIO	3380		
3880-4	258 N D,S	S 2	BL,CVC,EIO	3375		
3886	032 N D	D 2	BL,BY,CVC,CBY,EIO	3886		
3890	032 N D	D 2	BL,BY,CVC,CBY,EIO	3890		
3895	032 N D	D 2	BL,BY,CVC,CBY,EIO	3895		
3900	022 N D,S,S4	S 2	BL,BY,CNC,CVC,CBY,FCV,EIO	AFP1,AFP1-0,3900		
3935	022 N D,S,S4	S 2	BL,BY,CNC,CVC,CBY,FCV,EIO	AFP1-0,3935		
3990	002 Y S,S4	S 2	BL,CNC,CVC,ISD,DSD,FCV,EIO	3380,3390		
3990-1	002 N S	S 2	BL,CVC,EIO	3380		
3990-2	002 Y S,S4	S 2	BL,CNC,CVC,ISD,DSD,FCV,EIO	3380,3390		
3990-3	002 Y S,S4	S 2	BL,CNC,CVC,FCV,EIO	3380,3390		
3990-6	002 Y S,S4	S 2	BL,CNC,CVC,FCV,EIO	3380,3390		
3995	053 Y D,S,S4	S4 2	BL,CNC,CVC,FCV,EIO	3995		
3995-SDA	053 Y D,S,S4	S4 2	BL,CNC,CVC,FCV,EIO	3995		
3995-151	002 Y S,S4	S 2	BL,CNC,CVC,FCV,EIO	3995-151		
3995-153	002 Y S,S4	S 2	BL,CNC,CVC,FCV,EIO	3995-153		

Figure 200. Supported Hardware Report (Part 10 of 30)

		SUPPORTED HARDWARE - CONTROL UNITS (1)				TIME: 14:21	DATE: 2016-05-11	PAGE X- 11
TYPE-MODEL	UIM U	PROTCL	DP	IO	SUPPORTED CHPID TYPES	ATTACHABLE DEVICES		
4000	022 N D,S,S4	S	S	2	BL,BY,CNC,CVC,CBY,FCV,EIO	AFP1-0,4000		
4100	022 N D,S,S4	S	S	2	BL,BY,CNC,CVC,CBY,FC,FCV,EIO	AFP1-0,4100		
4245	012 N D	D	D	2	BL,BY,CVC,CBY,EIO	4245		
4248	012 N D	D	D	2	BL,BY,CVC,CBY,EIO	3262,4248,6262		
4370	022 N D,S,S4	S	S	2	BL,BY,CNC,CVC,CBY,FC,FCV,EIO	AFP1-0,4370		
5088-1	291 N D,S	D	D	2	BL,CVC,EIO	HFGD,3251,3279-2B,3279-3B,5080,5081,6090,6091		
5088-2	291 N D,S	D	D	2	BL,CVC,EIO	HFGD,3251,3279-2B,3279-3B,5080,5081,6090,6091		
6098	291 N S,S4	S	S	2	BL,CVC,EIO	HFGD,3251,3279-2B,3279-3B,5080,5081,6090,6091		
6120	308 N			2	IOC,EIO	3101,3151,3174,3178,3179,3180,3190,3191,3192,3193,3194,3262,3268,3270,3277,3278,3279,3286,3287,3289,3290,3812,3816,4224,4245,4250,5210		
6135	308 N			1	IOC,EIO	ICAE LAN		
6139	308 N			1	IOC,EIO	ICAILAN		
6140	308 N			1	IOC,EIO	ICAILAN		
6241	308 N			1	IOC,EIO	ICATELE2		
6242	308 N			1	IOC,EIO	ICATELE2		
6243	308 N			1	IOC,EIO	ICATELE2		
6244	308 N			1	IOC,EIO	ICATELE2		
6251	308 N			1	IOC,EIO	ICABSCA,ICAHDL,ICASDL,ICATELE2		
6252	308 N			1	IOC,EIO	ICABSCA,ICAHDL,ICASDL,ICATELE2		
6253	308 N			1	IOC,EIO	ICABSCA,ICAHDL,ICASDL,ICATELE2		
6254	308 N			1	IOC,EIO	ICABSCA,ICAHDL,ICASDL,ICATELE2		
6255	308 N			1	IOC,EIO	ICABSCA,ICAHDL,ICASDL,ICATELE2		
6256	308 N			1	IOC,EIO	ICABSCA,ICAHDL,ICASDL,ICATELE2		
6257	308 N			1	IOC,EIO	ICABSCA,ICAHDL,ICASDL,ICATELE2		
6258	308 N			1	IOC,EIO	ICABSCA,ICAHDL,ICASDL,ICATELE2		
6262	268 N D	D	D	2	BL,BY,CVC,CBY,EIO	6262		
6310	308 N			2	IOC,EIO	FB512,9332,9335,9336		
6311	308 N			2	IOC,EIO	9348		
7171	027 N D	D	D	2	BL,CVC,EIO	3161,3162,3163,3277,3278,3279,3286,3791L,7171		
7770-3	023 N D	D	D	2	BY,CBY	7770-3		
8232	014 N D,S,S4	S	S	2	BL,CVC,EIO	8232		
9032	051 Y			2	CNC	SWCH,9032		
9032-3	051 Y			2	CNC	SWCH,9032-3		
9032-5	051 Y			2	CNC,FCV	SWCH,9032-5		
9033	051 Y			2	CNC	SWCH,9033		
9340	002 Y S,S4	S	S	2	BL,CNC,CVC,FCV,EIO	9345		
9341	002 Y S,S4	S	S	2	BL,CNC,CVC,FCV,EIO	9345		
9343	002 Y S,S4	S	S	2	BL,CNC,CVC,FCV,EIO	9345		
9343-1	002 Y S,S4	S	S	2	BL,CNC,CVC,FCV,EIO	9345		

Figure 201. Supported Hardware Report (Part 11 of 30)

		SUPPORTED HARDWARE - CONTROL UNITS (1)				TIME: 14:21	DATE: 2016-05-11	PAGE X- 12
KEY	---	KEY DESCRIPTION						
TYPE-MODEL	---	SUPPORTED CONTROL UNIT TYPE						
UIM	---	INDEX OF UNIT INFORMATION MODULE SUPPORTING THIS CONTROL UNIT TYPE						
U	---	If Y, INDICATES THAT THE UNIT ADDRESS RANGE MUST START WITH 00 WHEN THE CONTROL UNIT IS CONNECTED TO AN ESCON CHANNEL PATH						
DP	---	DEFAULT PROTOCOL FOR PARALLEL CONTROL UNIT						
IO	---	DEFAULT I/O CONCURRENCY LEVEL FOR PARALLEL CONTROL UNIT						
PROTCL	---	PROTOCOL SUPPORTED BY THE CONTROL UNIT TYPE						
SUPPORTED CHPID TYPES	---	LIST OF CHANNEL PATH TYPES WHICH ARE SUPPORTED BY THIS CONTROL UNIT TYPE						
ATTACHABLE DEVICES	---	LIST OF ATTACHABLE DEVICE TYPES						

Figure 202. Supported Hardware Report (Part 12 of 30)

--CONTROL UNIT--		--ATTACHMENT COUNTS--			SUPPORTED HARDWARE - UNIT ADDRESSES--				CONTROL UNITS (2) - LOGICAL ADDRESSING--			TIME: 14:21 DATE: 2016-05-11			PAGE X- 13				
TYPE-MODEL	MD	CHP	MAX	DEV	MIN	MAX	REC	RANGE	LA	MIN	MAX	MULTIHOST	MAXCU	MAXPATH	MINGRP	SH	MXESC	MXFIC	DCM
AFP1	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
CTC	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
DUMMY	-	-	-	-	-	-	-	-	Y	0	F	N	-	-	-	-	-	-	NY
FCP	-	1	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
FCTC	-	1	-	-	-	-	-	-	Y	-	-	Y	-	-	-	-	-	-	NN
IQD	-	1	3	256	-	-	-	1	Y	-	-	Y	-	-	-	-	-	-	NN
NOCHECK	-	8	-	-	-	-	-	-	Y	00	FF	Y	-	-	-	-	-	-	YY
OSA	-	1	1	255	-	-	-	1	Y	-	-	Y	-	-	-	-	-	-	NN
OSC	-	1	-	254	-	-	-	1	N	-	-	-	-	-	-	-	-	-	NN
OSM	-	1	1	255	-	-	-	1	Y	-	-	Y	-	-	-	-	-	-	NN
OSN	-	1	-	255	-	-	-	1	N	-	-	-	-	-	-	-	-	-	NN
OSX	-	1	1	255	-	-	-	1	Y	-	-	Y	-	-	-	-	-	-	NN
RS6K	-	1	-	-	-	-	-	-	Y	-	-	Y	-	-	-	-	-	-	NN
RS6K-2	N	1	-	-	-	-	-	-	Y	-	-	Y	-	-	-	-	-	-	NN
SCTC	-	1	-	-	-	-	-	-	Y	-	-	Y	-	-	-	-	-	-	NN
SWCH	-	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
1287	-	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
1288	-	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
1750	-	-	-	256	-	-	-	-	Y	00	3F	Y	-	128	2	-	-	1024	YN
2032	-	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
2105	-	-	-	256	-	-	-	-	Y	00	FF	Y	-	256	2	-	64	256	YN
2107	-	-	-	256	-	-	-	-	Y	00	FE	Y	-	512	2	-	64	2048	YN
2107-FBA	N	-	-	256	-	-	-	-	Y	00	FE	Y	-	512	2	-	64	2048	YN
2501	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
2701	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
2710	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
2821	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
2835-2	N	-	-	2	-	-	16	-	N	-	-	-	-	-	-	-	-	-	NN
2840-2	N	-	-	4	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3088	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3160	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3170	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3172	-	2	-	-	-	-	-	-	Y	-	-	Y	-	-	-	-	-	-	NN
3174	-	-	-	256	-	-	32	-	Y	0	F	Y	8	-	-	-	-	-	NN
3203	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3258	-	-	-	16	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3262	-	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3272	-	-	-	32	-	-	32	-	N	-	-	-	-	-	-	-	-	-	NN
3274	-	-	-	32	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3300	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3380-CJ2	N	-	-	64	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3422	-	-	-	16	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3423	-	-	-	16	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3424	-	-	-	8	-	-	8	-	N	-	-	-	-	-	-	-	-	-	NN
3430	-	-	-	16	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3480	-	-	-	16	16	16	-	-	N	-	-	-	-	-	-	-	-	-	NN
3490	-	-	-	16	16	16	-	-	Y	00	10	Y	-	-	-	-	-	-	NN
3490-C1A	N	-	-	16	2	16	-	-	Y	00	10	Y	-	-	-	-	-	-	NN
3490-C10	N	-	-	16	2	16	-	-	Y	00	10	Y	-	-	-	-	-	-	NN
3490-C11	N	-	-	16	2	16	-	-	Y	00	10	Y	-	-	-	-	-	-	NN
3490-C2A	N	-	-	16	2	16	-	-	Y	00	10	Y	-	-	-	-	-	-	NN
3490-C22	N	-	-	16	2	16	-	-	Y	00	10	Y	-	-	-	-	-	-	NN
3505	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3540	-	-	-	2	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3590	-	-	-	16	1	16	-	-	Y	00	FF	Y	-	-	-	-	-	-	NN
3704	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3705	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3720	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3725	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3737	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3745	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3746	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3791L	-	-	-	256	-	-	32	-	Y	0	F	Y	8	-	-	-	-	-	NN
3800-1	Y	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3800-3	N	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3800-6	N	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3800-8	N	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3803	-	-	-	16	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3811	-	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3820	-	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3825	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3827	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN
3828	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	-	NN

Figure 203. Supported Hardware Report (Part 13 of 30)

--CONTROL UNIT--		--ATTACHMENT COUNTS--			SUPPORTED HARDWARE				- CONTROL UNITS (2)				TIME: 14:21 DATE: 2016-05-11				PAGE X- 14	
TYPE-MODEL	MD	CHP	MAX	DEV	MIN	MAX	REC	RANGE	LA	MIN	MAX	MULTIHOST	MAXCU	MAXPATH	MINGRP	SH	MXESC	MXFIC
3829	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3830-2	N	-	-	64	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3830-3	N	-	-	64	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3831	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3835	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3851	-	-	-	64	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3880-1	N	-	-	64	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3880-11	N	-	-	64	-	-	32	-	N	-	-	-	-	-	-	-	-	NN
3880-13	N	-	-	64	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3880-2	N	-	-	64	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3880-23	Y	-	-	64	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3880-3	N	-	-	64	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3880-4	N	-	-	64	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3886	-	-	-	2	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3890	-	-	-	2	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3895	-	-	-	2	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3900	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3935	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
3990	-	-	-	-	-	-	-	-	Y	0	F	Y	-	-	-	-	-	YN
3990-1	N	-	-	64	-	-	-	-	N	-	-	-	-	8	8	-	-	NN
3990-2	N	-	-	256	-	-	-	-	Y	0	F	Y	-	16	8	-	-	YN
3990-3	N	-	-	64	-	-	-	-	Y	0	F	Y	-	16	8	-	-	YN
3990-6	N	-	-	256	-	-	-	-	Y	0	F	Y	-	128	8	-	-	YN
3995	-	2	-	-	-	-	-	-	N	-	-	-	-	-	-	Y	-	NN
3995-SDA	N	2	-	256	2	256	16	1	N	-	-	-	-	-	-	Y	-	NN
3995-151	N	-	-	96	-	-	-	-	Y	0	F	Y	-	-	-	-	-	YN
3995-153	N	-	-	96	-	-	-	-	Y	0	F	Y	-	-	-	-	-	YN
4000	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
4100	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
4245	-	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
4248	-	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
4370	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
5088-1	N	-	-	16	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
5088-2	Y	-	-	32	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6098	-	-	-	192	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6120	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6135	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6139	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6140	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6241	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6242	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6243	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6244	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6251	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6252	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6253	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6254	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6255	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6256	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6257	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6258	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6262	-	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6310	-	-	-	64	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
6311	-	-	-	64	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
7171	-	-	-	64	-	-	32	-	N	-	-	-	-	-	-	-	-	NN
7770-3	N	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
8232	-	-	-	-	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
9032	-	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
9032-3	N	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
9032-5	N	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
9033	-	-	-	1	-	-	-	-	N	-	-	-	-	-	-	-	-	NN
9340	-	-	-	256	-	-	-	-	N	-	-	-	-	64	64	-	-	YN
9341	-	-	-	16	-	-	-	-	N	-	-	-	-	2	2	-	-	YN
9343	-	-	-	64	-	-	-	-	N	-	-	-	-	64	64	-	-	YN
9343-1	N	-	-	64	-	-	-	-	N	-	-	-	-	64	64	-	-	YN

Figure 204. Supported Hardware Report (Part 14 of 30)

```

KEY                KEY DESCRIPTION
---              -----
TYPE-MODEL        - SUPPORTED CONTROL UNIT TYPE
MD                - IF Y, MODEL IS DEFAULT
ATTACHMENT CNT   - MAXIMUM NUMBER OF CHANNEL PATHS THAT CAN BE CONNECTED TO C/U
  CHPMAX         - MINIMUM NUMBER OF DEVICES THAT MUST BE CONNECTED TO C/U
  DEVMIN         - MAXIMUM NUMBER OF DEVICES THAT CAN BE CONNECTED TO C/U
  DEVMAX         - MINIMUM NUMBER OF UNIT ADDRESSES THAT MUST BE DEFINED TO C/U
UNIT ADDRESSES   - MAXIMUM NUMBER OF UNIT ADDRESSES THAT CAN BE DEFINED TO C/U
  MIN           - RECOMMENDED NUMBER OF UNIT ADDRESSES
  MAX           - RANGE - MAXIMUM NUMBER OF UNIT ADDRESS RANGES THAT CAN BE DEFINED TO C/U
  REC           - IF Y, C/U SUPPORTS LOGICAL ADDRESSING (CUADD)
LOGICAL ADDRESSING LA - MINIMUM VALUE OF ALLOWED LOGICAL ADDRESS (CUADD)
  LA           - MAXIMUM VALUE OF ALLOWED LOGICAL ADDRESS (CUADD)
  MIN           - MULTHOST - IF Y, MULTIPLE HOSTS CAN CONNECT TO THE SAME LOGICAL ADDRESS (CUADD)
  MAX           - MAXXCU - MAXIMUM NUMBER OF LOGICAL CONTROL UNITS SUPPORTED (CUADD)
LOGICAL PATHS    - MAXIMUM NUMBER OF LOGICAL PATHS SUPPORTED BY C/U
  MAXPATH      - MINIMUM GROUP ATTACHMENT VALUE FOR LOGICAL PATHS
  MINGRP       - SH - IF Y, ONLY SINGLE HOST CAN ATTACH TO CONTROL UNIT AT A TIME
  SH           - MXESC - MAXIMUM NUMBER OF LOGICAL PATHS PER ESCON PORT
  MXESC        - MXFIC - MAXIMUM NUMBER OF LOGICAL PATHS PER FICON PORT
DCM              - DYNAMIC CHPID MANAGEMENT SUPPORT

Y                - CONTROL UNIT HAS THE CAPABILITY
N                - CONTROL UNIT DOES NOT HAVE THE CAPABILITY
-                - UIM DOES NOT DEFINE A VALUE OR VALUE IS NOT APPLICABLE

```

Figure 205. Supported Hardware Report (Part 15 of 30)

TYPE-MODEL	UIM	MVS	VM	MX	GR	RL	RH	SUPPORTED HARDWARE - DEVICES			RD	TM	ST	ATTACHABLE TO	CU
								RD	TM	ST					
AFPI-0	022	Y	Y	N	N	1	4095	1	Y	Y	AFPI, NOCHECK, 2710, 3160, 3170, 3300, 3800-3, 3800-6, 3800-8, 3820, 3825, 3827, 3828, 3829, 3831, 3835, 3900, 3935, 4000, 4100, 4370				
BCTC	014	Y	Y	N	N	1	4095	1	N	Y	NOCHECK, SCTC				
BSC1	026	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
BSC2	026	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
BSC3	026	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
CTC	014	Y	Y	N	N	1	4095	1	N	Y	CTC, NOCHECK, 3088, 3737				
CTCA	270	N	Y	N	N	1	4095	1	N	Y	CTC, NOCHECK, SCTC, 3088, 3737				
DUMMY	050	Y	Y	N	N	1	4095	1	Y	Y	DUMMY, NOCHECK				
FBA	002	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 2107-FBA				
FBASCSI	254	N	Y	N	N	1	4095	1	Y	Y					
FB512	258	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 6310				
FCP	254	N	Y	N	N	1	4095	1	N	N	FCP				
FCTC	014	Y	Y	N	N	1	4095	1	N	Y	FCTC, NOCHECK				
HFGD	291	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 5088-1, 5088-2, 6098				
ICABSCA	308	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 6251, 6252, 6253, 6254, 6255, 6256, 6257, 6258				
ICAELAN	308	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 6135				
ICAHDL	308	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 6251, 6252, 6253, 6254, 6255, 6256, 6257, 6258				
ICAILAN	308	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 6139, 6140				
ICASDL	308	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 6251, 6252, 6253, 6254, 6255, 6256, 6257, 6258				
ICATELEZ	308	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 6241, 6242, 6243, 6244, 6251, 6252, 6253, 6254, 6255, 6256, 6257, 6258				
IQD	058	Y	Y	N	N	3	4095	10	N	N	IQD				
OSA	058	Y	Y	N	N	1	4095	2	N	Y	OSA				
OSA-M	058	Y	Y	N	N	1	4095	2	N	Y	OSM				
OSA-X	058	Y	Y	N	N	1	4095	2	N	Y	OSX				
OSAD	058	Y	Y	N	N	1	4095	1	N	Y	OSA, OSM, OSN, OSX				
OSN	058	Y	Y	N	N	1	4095	1	N	N	OSN				
RS6K	056	Y	Y	N	N	1	4095	1	N	Y	NOCHECK, RS6K, RS6K-2				
SCTC	014	Y	Y	N	N	1	4095	1	N	Y	NOCHECK, SCTC				
SWCH	051	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, SWCH, 2032, 9032, 9032-3, 9032-5, 9033				
TWX	025	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
WTTA	025	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
1030	024	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
1050	024	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
1050X	024	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
115A	024	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
1287	032	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 1287				
1288	032	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 1288				
1403-N1	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2821				
1403-2	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2821				
1403-3	268	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2821				
1403-5	268	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2821				
1403-7	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2821				
2032	051	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2032				
2250-3	277	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2840-2, 3258				
2305-2	269	N	Y	Y	N	1	1	1	Y	Y	NOCHECK, 2835-2				
2501-B1	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2501				
2501-B2	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2501				
2540P-1	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2821				
2540R-1	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2821				
2701	279	N	Y	N	N	1	4095	1	Y	N	NOCHECK, 2701				
2710	022	Y	Y	N	N	1	4095	1	Y	Y	AFPI, NOCHECK, 2710				
2740	024	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
2740C	024	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
2740X	024	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
2741C	025	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
2741P	025	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746				
3088	014	Y	Y	N	N	1	4095	32	N	Y	NOCHECK, 3088				
3151	281	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3174, 3272, 3274, 3704, 3705, 3720, 3791L, 6120				
3160	022	Y	Y	N	N	1	4095	1	Y	Y	AFPI, NOCHECK, 3160				
3161	281	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 7171				
3162	281	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 7171				
3163	281	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 7171				
3167	281	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745				
3170	022	Y	Y	N	N	1	4095	1	Y	Y	AFPI, NOCHECK, 3170				
3172	057	Y	Y	N	N	1	4095	1	N	Y	NOCHECK, 3172				
3174	027	Y	Y	N	N	1	4095	1	Y	N	NOCHECK, 3174, 3704, 3705, 3720, 3725, 3745, 3746, 6120				
3178	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120				
3179	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120				
3180	260	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120				
3180-1	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120				
3190	260	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120				
3191	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120				
3192	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120				

Figure 206. Supported Hardware Report (Part 16 of 30)

TYPE-MODEL	UIM	MVS	VM	MX	GR	RL	RH	SUPPORTED HARDWARE - DEVICES			ATTACHABLE TO CU
								RD	TM	ST	
3192-F	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
3193	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
3194	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
3203-4	012	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 3203
3203-5	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3203
3211	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3811
3215	260	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, OSC
3216	278	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3811
3250	277	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3258
3251	277	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3258, 5088-1, 5088-2, 6098
3262-13	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 4248, 6120
3262-3	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 4248, 6120
3262-5	268	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3262, 3272, 3274, 3791L, 4248, 6120
3268-2	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
3270	260	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
3270-X	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, OSC, 3174, 3272, 3274, 3791L, 6120
3272	027	Y	N	N	N	1	4095	1	Y	N	NOCHECK, 3272, 3791L
3274	027	Y	Y	N	N	1	4095	1	Y	N	NOCHECK, 3272, 3274, 3704, 3705, 3720, 3725, 3745, 3746
3277-1	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3272, 3274, 3791L, 6120, 7171
3277-2	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3272, 3274, 3791L, 6120, 7171
3278-1	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3278-2	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3278-2A	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3278-3	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3278-4	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3278-5	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3279-S2B	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3279-S3G	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3279-2	260	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3279-2A	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3279-2B	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 5088-1, 5088-2, 6098, 6120, 7171
3279-2C	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3279-2X	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3279-3	260	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3279-3A	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3279-3B	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 5088-1, 5088-2, 6098, 6120, 7171
3279-3X	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120, 7171
3284-1	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3272, 3274, 3791L
3284-2	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3272, 3274, 3791L
3286-1	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, OSC, 3174, 3272, 3274, 3791L, 6120, 7171
3286-2	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, OSC, 3174, 3272, 3274, 3791L, 6120, 7171
3287-1	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, OSC, 3174, 3272, 3274, 3791L, 6120
3287-1C	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, OSC, 3174, 3272, 3274, 3791L, 6120
3287-2	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, OSC, 3174, 3272, 3274, 3791L, 6120
3287-2C	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, OSC, 3174, 3272, 3274, 3791L, 6120
3288	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3272, 3274, 3791L
3289-1	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
3289-2	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
3290	004	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
3300	022	Y	Y	N	N	1	4095	1	Y	Y	AFP1, NOCHECK, 3300
3330-1	257	N	Y	N	N	1	4095	2	Y	Y	NOCHECK, 3830-2, 3880-1, 3880-2
3330-11	257	N	Y	N	N	1	4095	2	Y	Y	NOCHECK, 3830-2, 3830-3, 3880-1, 3880-2
3330-2	257	N	Y	N	N	1	4095	2	Y	Y	NOCHECK, 3830-2, 3880-1, 3880-2
3330V	271	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3830-3
3333-1	257	N	Y	N	N	1	4095	2	Y	Y	NOCHECK, 3830-2, 3880-1, 3880-2
3333-11	257	N	Y	N	N	1	4095	2	Y	Y	NOCHECK, 3830-2, 3880-1, 3880-2
3340	257	N	Y	N	N	1	4095	2	Y	Y	NOCHECK, 3830-2, 3880-1, 3880-2
3344	257	N	Y	N	N	1	4095	2	Y	Y	NOCHECK, 3830-2, 3880-1
3350	257	N	Y	N	N	1	4095	2	Y	Y	NOCHECK, 3830-2, 3830-3, 3880-1, 3880-2
3370	258	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3880-1, 3880-11, 3880-2
3375	258	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3880-1, 3880-11, 3880-2, 3880-4
3380	002	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 1750, 2105, 2107, 3380-CJ2, 3880-13, 3880-2, 3880-23, 3880-3, 3990, 3990-1, 3990-2, 3990-3, 3990-6
3380-CJ2	002	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 1750, 2105, 2107, 3380-CJ2, 3880-13, 3880-2, 3880-23, 3880-3, 3990, 3990-1, 3990-2, 3990-3, 3990-6
3380A	002	Y	Y	Y	N	1	4095	1	Y	Y	NOCHECK, 1750, 2105, 2107
3380B	002	Y	Y	Y	N	1	4095	1	Y	Y	NOCHECK, 1750, 2105, 2107
3390	002	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 1750, 2105, 2107, 3990, 3990-2, 3990-3, 3990-6
3390A	002	Y	Y	Y	N	1	4095	1	Y	Y	NOCHECK, 1750, 2105, 2107
3390B	002	Y	Y	Y	N	1	4095	1	Y	Y	NOCHECK, 1750, 2105, 2107
3390D	002	Y	Y	Y	N	1	4095	1	Y	Y	NOCHECK, 1750, 2105, 2107
3390S	002	Y	Y	Y	N	1	4095	1	Y	Y	NOCHECK, 1750, 2105, 2107
3420-3	005	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3803
3420-4	005	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3803

Figure 207. Supported Hardware Report (Part 17 of 30)

TYPE-MODEL	UIM	MVS	VM	MX	GR	RL	RH	SUPPORTED HARDWARE - DEVICES			ATTACHABLE TO CU
								RD	TM	ST	
3420-5	005	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3803
3420-6	005	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3803
3420-7	005	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3803
3420-8	005	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3803
3422	005	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3422
3423	005	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 3423
3424	005	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3424
3430	005	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3430
3471	004	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3274
3472	004	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3274
3480	005	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3480
3481	004	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3274
3482	004	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3274
3483	004	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3274
3490	005	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3490, 3490-C1A, 3490-C10, 3490-C11, 3490-C2A, 3490-C22
3505	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3505
3525	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3505
3540	032	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3540
3590	005	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3590
3704	023	Y	Y	N	N	1	4095	1	Y	N	NOCHECK, 3704
3705	023	Y	Y	N	N	1	4095	1	Y	N	NOCHECK, 3705
3720	023	Y	Y	N	N	1	4095	1	Y	N	NOCHECK, 3720
3725	023	Y	Y	N	N	1	4095	1	Y	N	NOCHECK, 3725
3737	014	Y	Y	N	N	1	4095	1	N	Y	NOCHECK, 3737
3745	023	Y	Y	N	N	1	4095	1	Y	N	NOCHECK, OSN, 3745, 3746
3746	023	Y	Y	N	N	1	4095	1	Y	N	NOCHECK
3767-1	024	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746
3767-2	024	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746
3791L	027	Y	N	N	N	1	4095	1	Y	N	NOCHECK, 3174, 3272, 3274, 3704, 3705, 3720, 3725, 3745, 3746, 3791L, 7171
3800-1	011	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3800-1
3800-3	011	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3800-3
3800-6	011	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3800-6
3800-8	011	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3800-8
3812	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
3816	287	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
3820	022	Y	Y	N	N	1	4095	1	Y	N	NOCHECK, 3820
3825	022	Y	Y	N	N	1	4095	1	Y	Y	AFP1, NOCHECK, 3825
3827	022	Y	Y	N	N	1	4095	1	Y	Y	AFP1, NOCHECK, 3827
3828	022	Y	Y	N	N	1	4095	1	Y	Y	AFP1, NOCHECK, 3828
3829	022	Y	Y	N	N	1	4095	1	Y	Y	AFP1, NOCHECK, 3829
3831	022	Y	Y	N	N	1	4095	1	Y	Y	AFP1, NOCHECK, 3831
3835	022	Y	Y	N	N	1	4095	1	Y	Y	AFP1, NOCHECK, 3835
3851	271	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3851
3886	032	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3886
3890	032	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3890
3895	032	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3895
3900	022	Y	Y	N	N	1	4095	1	Y	Y	AFP1, NOCHECK, 3900
3935	022	Y	Y	N	N	1	4095	1	Y	Y	AFP1, NOCHECK, 3935
3995	053	Y	Y	N	Y	1	1	N	Y		NOCHECK, 3995, 3995-SDA
3995-SDA	053	Y	Y	N	N	1	256	16	N	Y	NOCHECK, 3995-SDA
3995-151	002	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3995-SDA, 3995-151
3995-153	002	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3995-SDA, 3995-153
4000	022	Y	Y	N	N	1	4095	1	Y	Y	AFP1, NOCHECK, 4000
4100	022	Y	N	N	N	1	4095	1	Y	Y	AFP1, NOCHECK, 4100
4224	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
4245	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 4245, 6120
4248	012	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 4248
4250	031	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
4370	022	Y	Y	N	N	1	4095	1	Y	Y	AFP1, NOCHECK, 4370
5080	291	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 5088-1, 5088-2, 6098
5081	291	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 5088-1, 5088-2, 6098
5210	031	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 3174, 3272, 3274, 3791L, 6120
6090	291	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 5088-1, 5088-2, 6098
6091	291	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 5088-1, 5088-2, 6098
6262	268	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 4248, 6262
7171	027	Y	Y	N	N	1	4095	1	Y	N	NOCHECK, 7171
7770-3	023	Y	N	N	N	1	4095	1	Y	N	NOCHECK, 7770-3
8232	014	Y	Y	N	N	1	4095	32	N	Y	NOCHECK, 8232
8383	025	Y	N	N	N	1	4095	1	Y	Y	NOCHECK, 2701, 3704, 3705, 3720, 3725, 3745, 3746
9032	051	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 9032
9032-3	051	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 9032, 9032-3
9032-5	051	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 9032, 9032-5
9033	051	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK, 9033
9332-40	258	N	Y	N	N	1	4095	1	Y	Y	NOCHECK, 6310

Figure 208. Supported Hardware Report (Part 18 of 30)

TYPE-MODEL	UIM	MVS	VM	MX	GR	RL	RH	SUPPORTED HARDWARE - DEVICES			ATTACHABLE TO CU
								RD	TM	ST	
9332-42	258	N	Y	N	N	1	4095	1	Y	Y	NOCHECK,6310
9332-60	258	N	Y	N	N	1	4095	1	Y	Y	NOCHECK,6310
9332-62	258	N	Y	N	N	1	4095	1	Y	Y	NOCHECK,6310
9335-B1	258	N	Y	N	N	1	4095	1	Y	Y	NOCHECK,6310
9336-10	258	N	Y	N	N	1	4095	1	Y	Y	NOCHECK,6310
9336-20	258	N	Y	N	N	1	4095	1	Y	Y	NOCHECK,6310
9345	002	Y	Y	N	N	1	4095	1	Y	Y	NOCHECK,9340,9341,9343,9343-1
9348-1	261	N	Y	N	N	1	4095	1	Y	Y	NOCHECK,6311

Figure 209. Supported Hardware Report (Part 19 of 30)

KEY	KEY DESCRIPTION
---	-----
TYPE-MODEL	- SUPPORTED DEVICE TYPE
UIM	- INDEX OF UNIT INFORMATION MODULE SUPPORTING THIS DEVICE TYPE
MVS	- DEVICE IS SUPPORTED FOR MVS DEFINITION
VM	- DEVICE TYPE IS SUPPORTED FOR VM DEFINITION
MX	- DEVICE IS A MULTI-EXPOSURE DEVICE OR A PARALLEL ACCESS VOLUME DEVICE
GR	- DEVICE IS A GROUP DEVICE
RL	- MINIMUM NUMBER OF DEVICES TO BE DEFINED
RH	- MAXIMUM NUMBER OF DEVICES TO BE DEFINED
RD	- DEFAULT NUMBER OF DEVICES TO BE DEFINED
TM	- DEFAULT TIMEOUT VALUE
ST	- DEFAULT STADET VALUE
ATTACHABLE TO CU	- LIST OF CONTROL UNIT TYPES TO WHICH DEVICE TYPE IS ATTACHABLE
Y	- DEVICE TYPE HAS THE CAPABILITY
N	- CAPABILITY IS NOT AVAILABLE

Figure 210. Supported Hardware Report (Part 20 of 30)

TYPE-MODEL	UIM GENERIC	DPREF	DYN	4DIG	UCB	NIP	SUPPORTED PARAMETERS (VALUES) / FEATURES
AFP1-0	022 AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER
BCTC	014 BCTC	8350	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
BSC1	026 AAA9	6900	N	N	N	N	(R)ADAPTER(BSCA),FEATURE,OFFLINE,(R)TCU(2701,2702,2703) / DUALCODE,DUALCOMM
BSC2	026 AAAA	7000	N	N	N	N	(R)ADAPTER(BSCA),FEATURE,OFFLINE,(R)TCU(2701,2702,2703) / AUTOANSR,AUTOCALL,DUALCODE,DUALCOMM
BSC3	026 AAAB	7100	N	N	N	N	(R)ADAPTER(BSCA),FEATURE,OFFLINE,(R)TCU(2701,2702,2703) / AUTOPOLL,DUALCODE,DUALCOMM
CTC	014 CTC	8400	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / 370
DUMMY	050 DUMMY	99991	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
FBA	002 FBA	260	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY /
FCTC	014 FCTC	8301	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
IQD	058 IQD	8362	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
OSA	058 OSA	8360	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
OSA-M	058 OSAM	8365	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
OSA-X	058 OSAX	8364	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
OSAD	058 OSAD	8361	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
OSN	058 OSN	8363	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
RS6K	056 RS6K	8389	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
SCTC	014 SCTC	8300	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
SWCH	051 SWCH	10500	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
TWX	025 AAA7	6700	N	N	N	N	(R)ADAPTER(TELE2),FEATURE,OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703) / AUTOANSR,AUTOCALL
WTTA	025 AAAD	7300	N	N	N	N	(R)ADAPTER(TELEW),OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703)
1030	024 AAA1	6100	N	N	N	N	(R)ADAPTER(IBM2),FEATURE,OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703) / AUTOPOLL
1050	024 AAA2	6200	N	N	N	N	(R)ADAPTER(IBM1,IBMT),FEATURE,OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703) / AUTOPOLL,AUTOANSR,AUTOCALL
1050X	024 AAAG	9700	N	N	N	N	(R)ADAPTER(IBM1,IBMT),FEATURE,OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703) / AUTOANSR,AUTOCALL
115A	024 AAA5	6500	N	N	N	N	(R)ADAPTER(TELE1),OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703)
1287	032 1287	5000	N	N	N	N	OFFLINE
1288	032 1288	5100	N	N	N	N	OFFLINE
1403-N1	012 1403	2100	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / UNVCHSET
1403-2	012 1403	2100	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / UNVCHSET
1403-7	012 1403	2100	Y	Y	N	N	OFFLINE,DYNAMIC
2032	051 SWCH	10500	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
2501-B1	012 2501	2300	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / CARDIMAGE
2501-B2	012 2501	2300	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / CARDIMAGE
2540P-1	012 2540-2	2900	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / CARDIMAGE
2540R-1	012 2540	2800	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / CARDIMAGE
2710	022 AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER
2740	024 AAA8	6800	N	N	N	N	(R)ADAPTER(IBM1),FEATURE,OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703) / AUTOPOLL,AUTOANSR,AUTOCALL,CHECKING,INTERRUPT,OIU,SCONTROL,XCONTROL
2740C	024 AAAI	9900	N	N	N	N	(R)ADAPTER(IBM1),(R)FEATURE,OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703) / AUTOANSR,AUTOCALL,CHECKING
2740X	024 AAAH	9800	N	N	N	N	(R)ADAPTER(IBM1),(R)FEATURE,OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703) / AUTOANSR,AUTOCALL,CHECKING
2741C	025 AAAF	7500	N	N	N	N	(R)ADAPTER(IBM1),FEATURE,OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703) / AUTOANSR
2741P	025 AAAG	7400	N	N	N	N	(R)ADAPTER(IBM1),FEATURE,OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703) / AUTOANSR
3088	014 CTC	8400	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / 370
3160	022 AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER
3170	022 AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER
3172	057 3172	8398	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
3174	027 3174	7650	Y	Y	N	N	OFFLINE,DYNAMIC,OWNER(VTAM,OTHER)
3178	004 3277-2	3800	Y	Y	Y	Y	FEATURE,OFFLINE,DYNAMIC,LOCANY / ASCACHAR,ASCBCHAR,DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,AUDALRM,MAGCDRD,NUMLOCK,PTREAD,SELPEN,ASKY3277,DEKY3277,EBKY3277,OCKY3277,KB70KEY,KB78KEY,KB81KEY
3179	004 3277-2	3800	Y	Y	Y	Y	FEATURE,OFFLINE,DYNAMIC,LOCANY / ASCACHAR,ASCBCHAR,DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,AUDALRM,MAGCDRD,NUMLOCK,PTREAD,SELPEN,ASKY3277,DEKY3277,EBKY3277,OCKY3277,KB70KEY,KB78KEY,KB81KEY
3180-1	004 3277-2	3800	Y	Y	Y	Y	FEATURE,OFFLINE,DYNAMIC,LOCANY / ASCACHAR,ASCBCHAR,DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,AUDALRM,MAGCDRD,NUMLOCK,PTREAD,SELPEN,ASKY3277,DEKY3277,EBKY3277,OCKY3277,KB70KEY,KB78KEY,KB81KEY
3191	004 3277-2	3800	Y	Y	Y	Y	FEATURE,OFFLINE,DYNAMIC,LOCANY / ASCACHAR,ASCBCHAR,DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,AUDALRM,MAGCDRD,NUMLOCK,PTREAD,SELPEN,ASKY3277,DEKY3277,EBKY3277,OCKY3277,KB70KEY,KB78KEY,KB81KEY
3192	004 3277-2	3800	Y	Y	Y	Y	FEATURE,OFFLINE,DYNAMIC,LOCANY / ASCACHAR,ASCBCHAR,DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,AUDALRM,MAGCDRD,NUMLOCK,PTREAD,SELPEN,ASKY3277,DEKY3277,EBKY3277,OCKY3277,KB70KEY,KB78KEY,KB81KEY
3192-F	004 3277-2	3800	Y	Y	Y	Y	FEATURE,OFFLINE,DYNAMIC,LOCANY / ASCACHAR,ASCBCHAR,DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,AUDALRM,MAGCDRD,NUMLOCK,PTREAD,SELPEN,ASKY3277,DEKY3277,EBKY3277,OCKY3277,KB70KEY,KB78KEY,KB81KEY

Figure 211. Supported Hardware Report (Part 21 of 30)

TYPE-MODEL	UIM	GENERIC	DPREF	DYN	4DIG	UCB	NIP	SUPPORTED PARAMETERS (VALUES) / FEATURES
3193	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3194	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3203-4	012	3203	2000	Y	Y	N	N	OFFLINE, DYNAMIC
3203-5	012	3203	2000	Y	Y	N	N	OFFLINE, DYNAMIC
3211	012	3211	1900	Y	Y	N	N	OFFLINE, DYNAMIC
3262-13	031	3286-2	4400	N	N	N	N	FEATURE, OFFLINE / DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, PTREAD
3262-3	031	3286-2	4400	N	N	N	N	FEATURE, OFFLINE / DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, PTREAD
3268-2	031	3286-2	4400	N	N	N	N	FEATURE, OFFLINE / DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, PTREAD
3270-X	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3272	027	3791L	7700	Y	Y	N	N	OFFLINE, DYNAMIC, OWNER (VTAM, OTHER)
3274	027	3274	7675	Y	Y	N	N	OFFLINE, DYNAMIC, OWNER (VTAM, OTHER)
3277-1	004	3277-1	3700	Y	Y	Y	N	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3277-2	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3278-1	004	3277-1	3700	Y	Y	Y	N	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3278-2	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3278-2A	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3278-3	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3278-4	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3278-5	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3279-S2B	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3279-S3G	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3279-2A	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3279-2B	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3279-2C	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3279-2X	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3279-3A	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3279-3B	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3279-3X	004	3277-2	3800	Y	Y	Y	Y	FEATURE, OFFLINE, DYNAMIC, LOCANY / ASCACHAR, ASCBCHAR, DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, AUDALRM, MAGCDRD, NUMLOCK, PTREAD, SELPEN, ASKY3277, DEKY3277, EBKY3277, OCKY3277, KB70KEY, KB78KEY, KB81KEY
3284-1	031	3284-1	4100	N	N	N	N	FEATURE, OFFLINE / DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, PTREAD
3284-2	031	3284-2	4200	N	N	N	N	FEATURE, OFFLINE / DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, PTREAD
3286-1	031	3286-1	4300	N	N	N	N	FEATURE, OFFLINE / DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, PTREAD
3286-2	031	3286-2	4400	N	N	N	N	FEATURE, OFFLINE / DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, PTREAD
3287-1	031	3286-2	4400	N	N	N	N	FEATURE, OFFLINE / DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, PTREAD
3287-1C	031	3286-1	4300	N	N	N	N	FEATURE, OFFLINE / DOCHAR, FRCHAR, GRCHAR, KACHAR, UKCHAR, PTREAD

Figure 212. Supported Hardware Report (Part 22 of 30)

TYPE-MODEL	UIM	GENERIC	DPREF	DYN	4DIG	UCB	NIP	SUPPORTED PARAMETERS (VALUES) / FEATURES
3287-2	031	3286-2	4400	N	N	N	N	FEATURE,OFFLINE / DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,PTREAD
3287-2C	031	3286-2	4400	N	N	N	N	FEATURE,OFFLINE / DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,PTREAD
3288	031	3286-2	4400	N	N	N	N	FEATURE,OFFLINE / DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,PTREAD
3289-1	031	3286-2	4400	N	N	N	N	FEATURE,OFFLINE / DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,PTREAD
3289-2	031	3286-2	4400	N	N	N	N	FEATURE,OFFLINE / DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,PTREAD
3290	004	3277-2	3800	Y	Y	Y	Y	FEATURE,OFFLINE,DYNAMIC,LOCANY / ASCACHAR,ASCBCHAR,DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,AUDALRM,MAGCDRD,NUMLOCK,PTREAD,SELPEN,ASKY3277,DEKY3277,EBKY3277,OCKY3277,KB70KEY,KB78KEY,KB81KEY
3300	022	AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER
3380	002	3380	290	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / SHARED,SHAREDUP
3380-CJ2	002	3380	290	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / SHARED,SHAREDUP
3380A	002	3380	290	Y	Y	Y	N	PRIVATE: WLMPAV
3380B	002	3380	290	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY,PRIVATE: WLMPAV / SHARED,SHAREDUP
3390	002	3390	280	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / SHARED,SHAREDUP
3390A	002	3390	280	Y	Y	Y	N	PRIVATE: WLMPAV
3390B	002	3390	280	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY,PRIVATE: WLMPAV / SHARED,SHAREDUP
3390D	002	3390	280	Y	Y	Y	N	FEATURE,DYNAMIC,LOCANY,PRIVATE: WLMPAV / SHARED
3390S	002	3390	280	Y	Y	Y	N	FEATURE,DYNAMIC,LOCANY,PRIVATE: WLMPAV / SHARED
3420-3	005	3400-3	1210	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / DATACONV,DUALDENS,SHARABLE,7-TRACK,9-TRACK
3420-4	005	3400-5	1200	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / SHARABLE,9-TRACK,OPT1600
3420-5	005	3400-3	1210	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / DATACONV,DUALDENS,SHARABLE,7-TRACK,9-TRACK
3420-6	005	3400-5	1200	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / SHARABLE,9-TRACK,OPT1600
3420-7	005	3400-3	1210	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / DATACONV,DUALDENS,SHARABLE,7-TRACK,9-TRACK
3420-8	005	3400-5	1200	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / SHARABLE,9-TRACK,OPT1600
3422	005	3400-6	1220	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / SHARABLE,9-TRACK,OPT1600
3423	005	3423	1380	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY
3424	005	3400-6	1220	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / SHARABLE
3430	005	3400-6	1220	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / 9-TRACK,OPT1600
3471	004	3277-2	3800	Y	Y	Y	Y	FEATURE,OFFLINE,DYNAMIC,LOCANY / ASCACHAR,ASCBCHAR,DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,AUDALRM,MAGCDRD,NUMLOCK,PTREAD,SELPEN,ASKY3277,DEKY3277,EBKY3277,OCKY3277,KB70KEY,KB78KEY,KB81KEY
3472	004	3277-2	3800	Y	Y	Y	Y	FEATURE,OFFLINE,DYNAMIC,LOCANY / ASCACHAR,ASCBCHAR,DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,AUDALRM,MAGCDRD,NUMLOCK,PTREAD,SELPEN,ASKY3277,DEKY3277,EBKY3277,OCKY3277,KB70KEY,KB78KEY,KB81KEY
3480	005	3480	1100	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY,PRIVATE: LIBRARY,AUTOSWITCH,LIBRARY-ID,LIBPORT-ID,MTL / SHARABLE,COMPACT
3481	004	3277-2	3800	Y	Y	Y	Y	FEATURE,OFFLINE,DYNAMIC,LOCANY / ASCACHAR,ASCBCHAR,DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,AUDALRM,MAGCDRD,NUMLOCK,PTREAD,SELPEN,ASKY3277,DEKY3277,EBKY3277,OCKY3277,KB70KEY,KB78KEY,KB81KEY
3482	004	3277-2	3800	Y	Y	Y	Y	FEATURE,OFFLINE,DYNAMIC,LOCANY / ASCACHAR,ASCBCHAR,DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,AUDALRM,MAGCDRD,NUMLOCK,PTREAD,SELPEN,ASKY3277,DEKY3277,EBKY3277,OCKY3277,KB70KEY,KB78KEY,KB81KEY
3483	004	3277-2	3800	Y	Y	Y	Y	FEATURE,OFFLINE,DYNAMIC,LOCANY / ASCACHAR,ASCBCHAR,DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,AUDALRM,MAGCDRD,NUMLOCK,PTREAD,SELPEN,ASKY3277,DEKY3277,EBKY3277,OCKY3277,KB70KEY,KB78KEY,KB81KEY
3490	005	3490	1000	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY,PRIVATE: LIBRARY,AUTOSWITCH,LIBRARY-ID,LIBPORT-ID,MTL / SHARABLE,COMPACT
3505	012	3505	2400	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / CARDIMAGE
3525	012	3525	2500	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / CARDIMAGE,TWOLINE,MULTILINE
3540	032	3540	5600	N	N	N	N	OFFLINE
3590	005	3590-1	950	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY,PRIVATE: LIBRARY,AUTOSWITCH,LIBRARY-ID,LIBPORT-ID,MTL / SHARABLE,COMPACT
3704	023	3704	7800	Y	Y	N	N	ADAPTER(CA1),OFFLINE,DYNAMIC,OWNER(VTAM,OTHER)
3705	023	3705	7600	Y	Y	N	N	(R)ADAPTER(TYPE1,TYPE2,TYPE3,TYPE4),OFFLINE,DYNAMIC,OWNER(VTAM,OTHER)
3720	023	3720	7575	Y	Y	N	N	(R)ADAPTER(TYPE5),OFFLINE,DYNAMIC,OWNER(VTAM,OTHER)
3725	023	3725	7550	Y	Y	N	N	(R)ADAPTER(TYPE5),OFFLINE,DYNAMIC,OWNER(VTAM,OTHER)
3737	014	CTC	8400	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / 370
3745	023	3745	7450	Y	Y	N	N	(R)ADAPTER(TYPE6,TYPE7),OFFLINE,DYNAMIC,OWNER(VTAM,OTHER)
3746	023	3746	7551	Y	Y	N	N	(R)ADAPTER(TYPE6,TYPE7),OFFLINE,DYNAMIC,OWNER(VTAM,OTHER)
3767-1	024	AA48	6800	N	N	N	N	(R)ADAPTER(IBM1),FEATURE,OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703) / AUTOPOLL,AUTOANSR,AUTOCALL,CHECKING,INTERRUPT,OIU,SCONTROL,XCONTROL
3767-2	024	AA48	6800	N	N	N	N	(R)ADAPTER(IBM1),FEATURE,OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703) / AUTOPOLL,AUTOANSR,AUTOCALL,CHECKING,INTERRUPT,OIU,SCONTROL,XCONTROL
3791L	027	3791L	7700	Y	Y	N	N	OFFLINE,DYNAMIC,OWNER(VTAM,OTHER)
3800-1	011	3800	1780	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER,CGS1,CGS2
3800-3	011	3800	1780	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER
3800-6	011	3800	1780	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER,CGS1,CGS2
3800-8	011	3800	1780	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER
3812	031	3286-2	4400	N	N	N	N	FEATURE,OFFLINE / DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,PTREAD
3820	022	3820	1800	Y	Y	N	N	OFFLINE,DYNAMIC
3825	022	AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER
3827	022	AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER
3828	022	AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER
3829	022	AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER

Figure 213. Supported Hardware Report (Part 23 of 30)

SUPPORTED HARDWARE - MVS DEVICES										TIME: 14:21	DATE: 2016-05-11	PAGE X- 24
TYPE-MODEL	UIM	GENERIC	DPREF	DYN	4DIG	UCB	NIP	SUPPORTED PARAMETERS (VALUES) / FEATURES				
3831	022	AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER				
3835	022	AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER				
3886	032	3886	4900	N	N	N	N	OFFLINE				
3890	032	3890	4800	N	N	N	N	OFFLINE				
3895	032	3895	5400	N	N	N	N	OFFLINE				
3900	022	AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER				
3935	022	AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER				
3995	053	3995	8600	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY				
3995-SDA	053	3995	8600	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY				
3995-151	002	3390	280	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / SHARED,SHAREDUP				
3995-153	002	3390	280	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / SHARED,SHAREDUP				
4000	022	AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER				
4100	022	AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER				
4224	031	3286-2	4400	N	N	N	N	FEATURE,OFFLINE / DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,PTREAD				
4245	012	4245	1890	Y	Y	N	N	OFFLINE,DYNAMIC				
4248	012	4248	1850	Y	Y	N	N	OFFLINE,DYNAMIC				
4250	031	3286-2	4400	N	N	N	N	FEATURE,OFFLINE / DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,PTREAD				
4370	022	AFP1	1750	Y	Y	N	N	FEATURE,OFFLINE,DYNAMIC / BURSTER				
5210	031	3286-2	4400	N	N	N	N	FEATURE,OFFLINE / DOCHAR,FRCHAR,GRCHAR,KACHAR,UKCHAR,PTREAD				
7171	027	3791L	7700	Y	Y	N	N	OFFLINE,DYNAMIC,OWNER(VTAM,OTHER)				
7770-3	023	AAAC	7200	Y	Y	N	N	OFFLINE,DYNAMIC,OWNER(VTAM,OTHER)				
8232	014	CTC	8400	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / 370				
8383	025	AAA6	6600	N	N	N	N	(R)ADAPTER(TELE1),OFFLINE,SETADDR(0,1,2,3),(R)TCU(2701,2702,2703)				
9032	051	SWCH	10500	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY				
9032-3	051	SWCH	10500	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY				
9032-5	051	SWCH	10500	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY				
9033	051	SWCH	10500	Y	Y	Y	N	OFFLINE,DYNAMIC,LOCANY				
9345	002	9345	270	Y	Y	Y	N	FEATURE,OFFLINE,DYNAMIC,LOCANY / SHARED,SHAREDUP				

Figure 214. Supported Hardware Report (Part 24 of 30)

SUPPORTED HARDWARE - MVS DEVICES		TIME: 14:21	DATE: 2016-05-11	PAGE X- 25
KEY	KEY DESCRIPTION			
---	-----			
TYPE-MODEL	- SUPPORTED DEVICE TYPE			
UIM	- INDEX OF UNIT INFORMATION MODULE SUPPORTING THIS DEVICE TYPE			
GENERIC	- GENERIC DEVICE TYPE			
DPREF	- DEFAULT DEVICE PREFERENCE VALUE			
DYN	- DEVICE SUPPORTS DYNAMIC I/O RECONFIGURATION			
4DIG	- DEVICE TYPE SUPPORTS DEVICE NUMBERS GREATER THAN 0FFF			
UCB	- DEVICE TYPE SUPPORTS UCB BEING LOCATED ABOVE 16 MB			
NIP	- DEVICE TYPE SUPPORTED AS NIP CONSOLE			
SUPPORTED PARAMETERS (VALUES) / FEATURES	- LIST OF SUPPORTED PARAMETERS (WITH SELECTION VALUES) AND FEATURES			
Y	- DEVICE TYPE HAS THE CAPABILITY			
N	- CAPABILITY IS NOT AVAILABLE			
(R)	- PARAMETER IS REQUIRED			
PRIVATE:	- START OF PARAMETERS PRIVATE TO UIM			

Figure 215. Supported Hardware Report (Part 25 of 30)

TYPE-MODEL	UIM VM D/T	CONS	SUPPORTED PARAMETERS (VALUES) / FEATURES
AFP1-0	278 AFP1	N	OFFLINE,UIRATE,EQID
BCTC	270 CTCA	N	OFFLINE,UIRATE,EQID
BSC1	282 3705	N	(R) ADAPTER (BSCA),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
BSC2	282 3705	N	(R) ADAPTER (BSCA),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
BSC3	282 3705	N	(R) ADAPTER (BSCA),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
CTC	270 CTCA	N	OFFLINE,UIRATE,EQID
CTCA	270 CTCA	N	OFFLINE,UIRATE,EQID
DUMMY	306 DUMMY	N	(R) CLASS (DASD,TAPE,TERM,DISPLAY,RDR,PRT,PUN,SWITCH),OFFLINE,UIRATE,EQID / DPS,RESERVE
FBASCSI	254 FBASCSI	N	OFFLINE,UIRATE,EQID,(R)ATTR(1750,2105,2107,2145,XIV,SCSI),(R)FCPDEV,(R)WPN,(R)LUN,PREFPATH
FB512	258 FB-512	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID
FCP	254 FCP	N	OFFLINE,UIRATE,EQID
FCTC	270 CTCA	N	OFFLINE,UIRATE,EQID
HFGD	291 HFGD	N	OFFLINE,UIRATE,EQID
ICABSCA	308 ICA	N	(R) ADAPTER (BSCA),OFFLINE,UIRATE,EQID
ICAE LAN	308 ICA	N	(R) ADAPTER (ELAN),OFFLINE,UIRATE,EQID
ICAHDL C	308 ICA	N	(R) ADAPTER (HDL C),OFFLINE,UIRATE,EQID
ICAILAN	308 ICA	N	(R) ADAPTER (ILAN),OFFLINE,UIRATE,EQID
ICASDL C	308 ICA	N	(R) ADAPTER (SDLC),OFFLINE,UIRATE,EQID
ICATELE2	308 ICA	N	(R) ADAPTER (TELE2),OFFLINE,UIRATE,EQID
IQD	314 OSA	N	OFFLINE,UIRATE,EQID
OSA	314 OSA	N	OFFLINE,UIRATE,EQID
OSA-M	314 OSA	N	OFFLINE,UIRATE,EQID
OSA-X	314 OSA	N	OFFLINE,UIRATE,EQID
OSAD	314 OSAD	N	OFFLINE,UIRATE,EQID
OSN	314 OSN	N	OFFLINE,UIRATE,EQID
RS6K	312 3088	N	OFFLINE,UIRATE,EQID
SCTC	270 CTCA	N	OFFLINE,UIRATE,EQID
SWCH	307 9032	N	OFFLINE,UIRATE,EQID
TWX	281 3705	N	(R) ADAPTER (TELE2),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
1050	280 3705	N	(R) ADAPTER (IBM1),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
1050X	280 3705	N	(R) ADAPTER (IBM1),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
1287	288 1287	N	OFFLINE,UIRATE,EQID
1288	288 1288	N	OFFLINE,UIRATE,EQID
1403-N1	268 1403	N	AFP,CHARS,CLASS,DEST,FOLD,FORM,IMAGE,OFFLINE,SEP,UIRATE,EQID / UNVCHSET
1403-2	268 1403	N	AFP,CHARS,CLASS,DEST,FOLD,FORM,IMAGE,OFFLINE,SEP,UIRATE,EQID / UNVCHSET
1403-3	268 1403	N	AFP,CHARS,CLASS,DEST,FOLD,FORM,IMAGE,OFFLINE,SEP,UIRATE,EQID / UNVCHSET
1403-5	268 1403	N	AFP,CHARS,CLASS,DEST,FOLD,FORM,IMAGE,OFFLINE,SEP,UIRATE,EQID / UNVCHSET
1403-7	268 1403	N	AFP,CHARS,CLASS,DEST,FOLD,FORM,IMAGE,OFFLINE,SEP,UIRATE,EQID / UNVCHSET
2032	307 2032	N	OFFLINE,UIRATE,EQID
2250-3	277 2250	N	OFFLINE,UIRATE,EQID
2305-2	269 2305-2	N	OFFLINE,SHARED,UIRATE,EQID
2501-B1	266 2501	N	CLASS,OFFLINE,UIRATE,EQID
2501-B2	266 2501	N	CLASS,OFFLINE,UIRATE,EQID
2540P-1	268 2540P	N	CLASS,FORM,OFFLINE,SEP,UIRATE,EQID
2540R-1	266 2540R	N	CLASS,OFFLINE,UIRATE,EQID
2701	279 2701	N	(R) ADAPTER (BSCA,IBM1,TELE2),OFFLINE,UIRATE,EQID
2710	278 AFP1	N	OFFLINE,UIRATE,EQID
2740	280 3705	N	(R) ADAPTER (IBM1),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
2740C	280 3705	N	(R) ADAPTER (IBM1),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
2740X	280 3705	N	(R) ADAPTER (IBM1),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
2741C	281 3705	N	(R) ADAPTER (IBM1),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
2741P	281 3705	N	(R) ADAPTER (IBM1),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
3088	270 3088	N	OFFLINE,UIRATE,EQID
3151	281 3705	N	(R) ADAPTER (TELE2),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
3160	278 AFP1	N	OFFLINE,UIRATE,EQID
3161	281 3705	N	(R) ADAPTER (TELE2),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
3162	281 3705	N	(R) ADAPTER (TELE2),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
3163	281 3705	N	(R) ADAPTER (TELE2),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
3167	281 3705	N	(R) ADAPTER (TELE2),BASEADD,OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID
3170	278 AFP1	N	OFFLINE,UIRATE,EQID
3172	313 3172	N	OFFLINE,UIRATE,EQID
3174	283 3705	N	(R) ADAPTER (TYPE4),CPNAME,(R)MODEL(A1,A2,B1,B2,B3,B4,C1,C2,C3,C4,C5,C6,D1,D2,D3,D4,D5,D6,D7,D8,E1,E2,E3,E4,E5,E6,E7,E8,F1,F2,F3,F4,F5,F6,F7,F8,G1,G2,G3,G4,G5,G6,G7,G8,H1,H2,H3,H4,H5,H6,H7,H8,J1,J2,J3,J4,K1,K2,K3,K4,L1,L2,L3,L4,1,2,3,4,5,6,7,8),OFFLINE,UIRATE,EQID
3178	260 3178	Y	MODEL(2,3,4,5),OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3179	260 3179	Y	MODEL(2,3),OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3180	260 3180	Y	MODEL(2,3,4,5),OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3180-1	260 3180	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3190	260 3190	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3191	260 3190	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3192	260 3190	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3192-F	260 3190	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3193	260 3190	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR

Figure 216. Supported Hardware Report (Part 26 of 30)

TYPE-MODEL	UIM	VM D/T	CONS	SUPPORTED PARAMETERS (VALUES) / FEATURES
3194	260	3190	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3203-5	268	3203-5	N	AFP,CHARS,CLASS,DEST,FCB,FOLD,FORM,IMAGE,OFFLINE,SEP,UIRATE,LIMIT,EQID / UNVCHSET
3211	268	3211	N	AFP,CHARS,CLASS,DEST,FCB,FOLD,FORM,IMAGE,INDEX,OFFLINE,SEP,UIRATE,LIMIT,EQID / UNVCHSET
3215	260	3215	Y	OFFLINE,UIRATE,EQID
3216	278	AFP1	N	OFFLINE,UIRATE,EQID
3250	277	3250	N	OFFLINE,UIRATE,EQID
3251	277	3250	N	OFFLINE,UIRATE,EQID
3262-13	287	3287	N	OFFLINE,UIRATE,EQID
3262-3	287	3287	N	OFFLINE,UIRATE,EQID
3262-5	268	3262	N	AFP,CHARS,CLASS,DEST,FCB,FOLD,FORM,IMAGE,OFFLINE,SEP,UIRATE,LIMIT,EQID
3268-2	287	3287	N	OFFLINE,UIRATE,EQID
3270	260	3270	Y	MODEL(2A,2C,2,3,4,5),OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3270-X	260	3270	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3274	283	3705	N	(R)ADAPTER(TYPE4),CPNAME,(R)MODEL(A1,A2,B1,B2,B3,B4,C1,C2,C3,C4,C5,C6,D1,D2,D3,D4,D5,D6,D7,D8,E1,E2,E3,E4,E5,E6,E7,E8,F1,F2,F3,F4,F5,F6,F7,F8,G1,G2,G3,G4,G5,G6,G7,G8,H1,H2,H3,H4,H5,H6,H7,H8,J1,J2,J3,J4,K1,K2,K3,K4,L1,L2,L3,L4,1,2,3,4,5,6,7,8),OFFLINE,UIRATE,EQID
3277-1	260	3277	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3277-2	260	3277	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3278-1	260	3278	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3278-2	260	3278	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3278-2A	260	3278-2A	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3278-3	260	3278-3	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3278-4	260	3278-4	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3278-5	260	3278-5	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3279-S2B	260	3279	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3279-S3G	260	3279	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3279-2	260	3279	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3279-2A	260	3279	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3279-2B	260	3279	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3279-2C	260	3279-2C	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3279-2X	260	3279	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3279-3	260	3279-3	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3279-3A	260	3279	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3279-3B	260	3279	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3279-3X	260	3279	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3284-1	287	3284	N	OFFLINE,UIRATE,EQID
3284-2	287	3284	N	OFFLINE,UIRATE,EQID
3286-1	287	3286	N	OFFLINE,UIRATE,EQID
3286-2	287	3286	N	OFFLINE,UIRATE,EQID
3287-1	287	3287	N	OFFLINE,UIRATE,EQID
3287-1C	287	3287	N	OFFLINE,UIRATE,EQID
3287-2	287	3287	N	OFFLINE,UIRATE,EQID
3287-2C	287	3287	N	OFFLINE,UIRATE,EQID
3288	287	3288	N	OFFLINE,UIRATE,EQID
3289-1	287	3289	N	OFFLINE,UIRATE,EQID
3289-2	287	3289	N	OFFLINE,UIRATE,EQID
3290	260	3290	Y	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR
3300	278	AFP1	N	OFFLINE,UIRATE,EQID
3330-1	257	3330-1	N	OFFLINE,SHARED,UIRATE,EQID
3330-11	257	3330-11	N	OFFLINE,SHARED,UIRATE,EQID
3330-2	257	3330-2	N	OFFLINE,SHARED,UIRATE,EQID
3330V	271	3330	N	OFFLINE,SHARED,UIRATE,EQID
3333-1	257	3330-1	N	OFFLINE,SHARED,UIRATE,EQID
3333-11	257	3330-11	N	OFFLINE,SHARED,UIRATE,EQID
3340	257	3340	N	OFFLINE,SHARED,UIRATE,EQID
3344	257	3340	N	OFFLINE,SHARED,UIRATE,EQID
3350	257	3350	N	OFFLINE,SHARED,UIRATE,EQID
3370	258	3370	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID
3375	258	3375	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID
3380	258	3380	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID
3380-CJ2	258	3380	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID
3380A	258	3380A	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID
3380B	258	3380B	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID
3390	258	3390	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID
3390A	258	3390A	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID
3390B	258	3390B	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID
3420-3	261	3420-3	N	OFFLINE,UIRATE,EQID / CONV,DUALDENS,TRANS,7-TRACK
3420-4	261	3420-4	N	OFFLINE,UIRATE,EQID / CONV,DUALDENS,TRANS,7-TRACK
3420-5	261	3420-5	N	OFFLINE,UIRATE,EQID / CONV,DUALDENS,TRANS,7-TRACK
3420-6	261	3420-6	N	OFFLINE,UIRATE,EQID / CONV,DUALDENS,TRANS,7-TRACK
3420-7	261	3420-7	N	OFFLINE,UIRATE,EQID / CONV,DUALDENS,TRANS,7-TRACK
3420-8	261	3420-8	N	OFFLINE,UIRATE,EQID / CONV,DUALDENS,TRANS,7-TRACK
3422	261	3422	N	OFFLINE,UIRATE,EQID
3424	261	3424	N	OFFLINE,UIRATE,EQID

Figure 217. Supported Hardware Report (Part 27 of 30)

				SUPPORTED HARDWARE - VM DEVICES	TIME: 14:21 DATE: 2016-05-11 PAGE X- 28
TYPE-MODEL	UIM VM D/T	CONS	SUPPORTED PARAMETERS (VALUES) / FEATURES		
3430	261 3430	N	OFFLINE,UIRATE,EQID		
3480	261 3480	N	OFFLINE,UIRATE,EQID		
3490	261 3490	N	OFFLINE,UIRATE,EQID		
3505	266 3505	N	CLASS,OFFLINE,UIRATE,EQID		
3525	268 3525	N	CLASS,FORM,OFFLINE,UIRATE,EQID		
3540	288 3540	N	OFFLINE,UIRATE,EQID		
3590	261 3590	N	OFFLINE,UIRATE,EQID		
3704	279 3704	N	(R)ADAPTER(BSCA,IBM1,TELE2,TYPE1),BASEADD,CPNAME,MODEL(A1,A2,A3,A4,1,2,3,4),OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID		
3705	279 3705	N	(R)ADAPTER(BSCA,IBM1,TELE2,TYPE4),BASEADD,CPNAME,MODEL(A1,A2,B1,B2,B3,B4,C1,C2,C3,C4,C5,C6,D1,D2,D3,D4,D5,D6,D7,D8,E1,E2,E3,E4,E5,E6,E7,E8,F1,F2,F3,F4,F5,F6,F7,F8,G1,G2,G3,G4,G5,G6,G7,G8,H1,H2,H3,H4,H5,H6,H7,H8,J1,J2,J3,J4,K1,K2,K3,K4,L1,L2,L3,L4,1,2,3,4,5,6,7,8),OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID		
3720	279 3705	N	(R)ADAPTER(TYPE4),CPNAME,(R)MODEL(A1,A2,B1,B2,B3,B4,C1,C2,C3,C4,C5,C6,D1,D2,D3,D4,D5,D6,D7,D8,E1,E2,E3,E4,E5,E6,E7,E8,F1,F2,F3,F4,F5,F6,F7,F8,G1,G2,G3,G4,G5,G6,G7,G8,H1,H2,H3,H4,H5,H6,H7,H8,J1,J2,J3,J4,K1,K2,K3,K4,L1,L2,L3,L4,1,2,3,4,5,6,7,8),OFFLINE,UIRATE,EQID		
3725	279 3705	N	(R)ADAPTER(BSCA,IBM1,TELE2,TYPE4),BASEADD,CPNAME,MODEL(A1,A2,B1,B2,B3,B4,C1,C2,C3,C4,C5,C6,D1,D2,D3,D4,D5,D6,D7,D8,E1,E2,E3,E4,E5,E6,E7,E8,F1,F2,F3,F4,F5,F6,F7,F8,G1,G2,G3,G4,G5,G6,G7,G8,H1,H2,H3,H4,H5,H6,H7,H8,J1,J2,J3,J4,K1,K2,K3,K4,L1,L2,L3,L4,1,2,3,4,5,6,7,8),OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID		
3737	270 CTCA	N	OFFLINE,UIRATE,EQID		
3745	279 3705	N	(R)ADAPTER(BSCA,IBM1,TELE2,TYPE4),BASEADD,CPNAME,MODEL(A1,A2,B1,B2,B3,B4,C1,C2,C3,C4,C5,C6,D1,D2,D3,D4,D5,D6,D7,D8,E1,E2,E3,E4,E5,E6,E7,E8,F1,F2,F3,F4,F5,F6,F7,F8,G1,G2,G3,G4,G5,G6,G7,G8,H1,H2,H3,H4,H5,H6,H7,H8,J1,J2,J3,J4,K1,K2,K3,K4,L1,L2,L3,L4,1,2,3,4,5,6,7,8),OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID		
3746	279 3705	N	(R)ADAPTER(BSCA,IBM1,TELE2,TYPE4),BASEADD,CPNAME,MODEL(A1,A2,B1,B2,B3,B4,C1,C2,C3,C4,C5,C6,D1,D2,D3,D4,D5,D6,D7,D8,E1,E2,E3,E4,E5,E6,E7,E8,F1,F2,F3,F4,F5,F6,F7,F8,G1,G2,G3,G4,G5,G6,G7,G8,H1,H2,H3,H4,H5,H6,H7,H8,J1,J2,J3,J4,K1,K2,K3,K4,L1,L2,L3,L4,1,2,3,4,5,6,7,8),OFFLINE,SETADDR(0,1,2,3,4),UIRATE,EQID		
3800-1	267 3800	N	AFP,CHARS,CLASS,DEST,DPMSIZE(1,2,3,4,5,6,7,8,9),FCB,FLASH,FORM,IMAGE,OFFLINE,SEP,UIRATE,MARK,LIMIT,EQID / 4WCGMS		
3800-3	267 3800-3	N	AFP,CHARS,CLASS,DEST,DPMSIZE(1,2,3,4,5,6,7,8,9),FCB,FLASH,FORM,IMAGE,OFFLINE,SEP,UIRATE,MARK,LIMIT,EQID / 4WCGMS		
3800-6	267 3800-6	N	AFP,CHARS,CLASS,DEST,DPMSIZE(1,2,3,4,5,6,7,8,9),FCB,FLASH,FORM,IMAGE,OFFLINE,SEP,UIRATE,MARK,LIMIT,EQID / 4WCGMS		
3800-8	267 3800-8	N	AFP,OFFLINE,UIRATE,EQID / 4WCGMS		
3812	287 3287	N	OFFLINE,UIRATE,EQID		
3816	287 3287	N	OFFLINE,UIRATE,EQID		
3820	278 3820	N	OFFLINE,UIRATE,EQID		
3825	278 AFP1	N	OFFLINE,UIRATE,EQID		
3827	278 AFP1	N	OFFLINE,UIRATE,EQID		
3828	278 AFP1	N	OFFLINE,UIRATE,EQID		
3829	278 AFP1	N	OFFLINE,UIRATE,EQID		
3831	278 AFP1	N	OFFLINE,UIRATE,EQID		
3835	278 AFP1	N	OFFLINE,UIRATE,EQID		
3851	271 3851	N	OFFLINE,SHARED,UIRATE,EQID		
3886	288 3886	N	OFFLINE,UIRATE,EQID		
3890	288 3890	N	OFFLINE,UIRATE,EQID		
3895	288 3895	N	OFFLINE,UIRATE,EQID		
3900	278 AFP1	N	OFFLINE,UIRATE,EQID		
3935	278 AFP1	N	OFFLINE,UIRATE,EQID		
3995	309 3088	N	OFFLINE,UIRATE,EQID		
3995-SDA	309 3088	N	OFFLINE,UIRATE,EQID		
3995-151	258 3390	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID		
3995-153	258 3390	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID		
4000	278 AFP1	N	OFFLINE,UIRATE,EQID		
4224	287 3287	N	OFFLINE,UIRATE,EQID		
4245	268 4245	N	AFP,CHARS,CLASS,DEST,FCB,FOLD,FORM,IMAGE,OFFLINE,SEP,UIRATE,LIMIT,EQID		
4248	268 4248	N	AFP,CHARS,CLASS,DEST,FCB,FOLD,FORM,IMAGE,OFFLINE,SEP,UIRATE,LIMIT,EQID		
4370	278 AFP1	N	OFFLINE,UIRATE,EQID		
5080	291 5080	N	OFFLINE,UIRATE,EQID		
5081	291 5080	N	OFFLINE,UIRATE,EQID		
5210	287 3287	N	OFFLINE,UIRATE,EQID		
6090	291 5080	N	OFFLINE,UIRATE,EQID		
6091	291 5080	N	OFFLINE,UIRATE,EQID		
6262	268 4248	N	AFP,CHARS,CLASS,DEST,FCB,FOLD,FORM,IMAGE,OFFLINE,SEP,UIRATE,LIMIT,EQID		
7171	283 3278	N	OFFLINE,UIRATE,EQID / EMUL3270,E3270HLD,OPRDR		
8232	270 3088	N	OFFLINE,UIRATE,EQID		
9032	307 9032	N	OFFLINE,UIRATE,EQID		
9032-3	307 9032	N	OFFLINE,UIRATE,EQID		
9032-5	307 9032	N	OFFLINE,UIRATE,EQID		
9033	307 9033	N	OFFLINE,UIRATE,EQID		
9332-40	258 9332	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID		
9332-42	258 9332	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID		

Figure 218. Supported Hardware Report (Part 28 of 30)

				SUPPORTED HARDWARE - VM DEVICES	TIME: 14:21 DATE: 2016-05-11 PAGE X- 29
TYPE-MODEL	UIM VM D/T	CONS	SUPPORTED PARAMETERS (VALUES) / FEATURES		
9332-60	258 9332	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID		
9332-62	258 9332	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID		
9335-B1	258 9335	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID		
9336-10	258 9336	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID		
9336-20	258 9336	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID		
9345	258 9345	N	OFFLINE,SHARED,UIRATE,MDC(OFF,DFLTOFF,DFLTON),EQID		
9348-1	261 9348	N	OFFLINE,UIRATE,EQID / DUALDENS		

Figure 219. Supported Hardware Report (Part 29 of 30)

```

KEY                                SUPPORTED HARDWARE - VM DEVICES          TIME: 14:21  DATE: 2016-05-11  PAGE X- 30
---                                KEY DESCRIPTION
TYPE-MODEL                         - SUPPORTED DEVICE TYPE
UIM                                 - INDEX OF UNIT INFORMATION MODULE SUPPORTING THIS DEVICE TYPE
VM D/T                             - VM DEVICE TYPE
CONS                               - DEVICE TYPE IS SUPPORTED AS VM CONSOLE
SUPPORTED PARAMETERS (VALUES) / FEATURES - LIST OF SUPPORTED PARAMETERS (WITH SELECTION VALUES) AND FEATURES
Y                                  - DEVICE TYPE HAS THE CAPABILITY
N                                  - CAPABILITY IS NOT AVAILABLE
(R)                                - PARAMETER IS REQUIRED

```

Figure 220. Supported Hardware Report (Part 30 of 30)

I/O Definition Reference

```

I/O DEFINITION REFERENCE - DEVICES          TIME: 17:31  DATE: 2012-10-21  PAGE Y- 305
3490    TAPE    Magnetic Tape Subsystem          /    1
UNIT=3490

Physical Configuration Information

Physical Device Types

Physical Device
3490-B20
3490-B40
Attachable to the following control units
3490-A10
3490-A20

Physical Device
(Integrated Tape Subsystem)
3490-C10
3490-C11
3490-C22

Physical Device
(Integrated Tape Subsystem inside a 3494)
3490-C1A
3490-C2A

Physical Device
3494 and 3495 are tape libraries containing automation,
library manager, one or more tape control units, storage
cells and tape cartridges. To prepare the IODF, no need
to define 3494 and 3495 explicitly. However, need to
indicate the tape devices as library tape devices by
specifying LIBRARY=YES in the device definition.

Logical Configuration Rules

For Channel Subsystem
- Up to maximum 4 channel paths for A10, 8 for A20,
  2 for C1A or C2A.
- Range of 16 unit addresses for control unit. Also
  accept a minimum of 2 addresses for integrated tape
  subsystem.

For Operating System
- LIBRARY=YES, if devices are installed in a system-
  managed IBM 3494 or IBM 3495 Tape Library.
- LIBRARY=YES or NO for BTL managed library drives
- DYNAMIC=YES, if devices are dynamically reconfigured
- AUTOSWITCH=YES, if tape drives are dynamically switched
  between system (MVS/ESA 5.2.0)

Configuration Example
CHPID PATH=(22,27),TYPE=CNC

```

Figure 221. I/O Definition Reference (Part 1 of 5)

3490

TAPE Magnetic Tape Subsystem

/ 2

```

CNTLUNIT CUNUMBR=0501,PATH=(22,27),UNIT=3490, X
UNITADD=((00,16))
IODEVICE ADDRESS=(540,016),CUNUMBR=(0501),UNIT=3490, X
OFFLINE=YES,DYNAMIC=YES, X
USERPRM=(LIBRARY,YES),(AUTOSWITCH,YES))

```

Software Prerequisites

Minimum version and release to operate the device in
MVS/ESA environment:

- MVS/SP - JES2 3.1
- MVS/DFP 3.1

Minimum Product Levels

- DFSORT release 11 (release 12 for C1A,C2A)
- EREP 3.4 (3.5 for C1A,C2A)
- DFHSM 2.6
- DFDSS 2.5

References

- MVS/ESA & MVS/XA Support for 3490 Magnetic Tape
Subsystem. GC28-1141
- IBM 3490E Planning and Migration Guide GC35-0219
- IBM 3490 Planning and Migration Guide GC35-0116

Notes

- For the latest information, contact your local IBM
Marketing Representative.

-end-

Device number definition values: ADDRESS=(device number<,number-of-devices>)

device number: 1 - 4 hexadecimal number in the range 0000 - FFFF.

number-of-devices: Number of sequential device numbers to be assigned to the devices.

Minimum value: 1
Default value: 1
Maximum value: 4095

Channel Subsystem information:

When attached to a parallel interface:

I/O interface time out function default: TIMEOUT=YES
Status verification facility default: STADET=YES

Figure 222. I/O Definition Reference (Part 2 of 5)

3490 TAPE Magnetic Tape Subsystem / 3

Unit address: UNITADD=xx

The unit address is a hexadecimal value from 00 to FF which must be specified in the unit address range of the control unit.
Default unit address are the last 2 digits of the device number.

MVS configuration information:

Generic name: 3490 It may be modified dependent on the specified features.

Support of dynamic I/O reconfiguration: Yes

Support of device numbers greater than 0FFF: Yes

Support of UCBS above 16 MB storage: Yes

Required parameters: None.

Optional parameters:

OFFLINE Device considered online or offline at IPL
Default value: OFFLINE=NO

Specifies whether MVS is to consider the device online or offline at IPL.

Yes The device is considered offline at IPL.

No The device is considered online at IPL.
(Default)

If MVS needs the device during IPL, specify No.

DYNAMIC Device supports dynamic configuration

Specify yes to indicate that the device is to be eligible for Dynamic I/O Configuration.

LIBRARY Device supports auto tape library

Data type is YES or NO
Pre-selected: No

Specify YES to indicate that the device belongs to an automated tape library.

AUTOSWITCH Device is automatically switchable

Data type is YES or NO
Pre-selected: No

Specify YES to indicate that the device should

Figure 223. I/O Definition Reference (Part 3 of 5)

3490 TAPE Magnetic Tape Subsystem / 4

be treated as an automatically switchable device

Supported features:

ALTCTRL Separate physical control unit path

Specify Yes to indicate that there is a separate physical control unit path to the device.

SHARABLE Device is Sharable between systems

Specify Yes to indicate that the 3803 two-channel switch is used for partitioning and that magnetic tape drives can be shared between two processors.

Do not allocate or unload a shared tape drive.

If specify Yes for SHARABLE, HCD forces a value of Yes for the OFFLINE parameter, even if you specify No for OFFLINE.

COMPACT Compaction

Specify Yes to indicate that compaction is available for tape devices.

Compaction is a method of compressing and encoding data in order to reduce storage space.

Figure 224. I/O Definition Reference (Part 4 of 5)

INDEX

AFP1-0 Y- 1	3088 Y- 95	3279-2C Y- 209	3480 Y- 300
BCTC Y- 3	3151 Y- 97	3279-2X Y- 116	3481 Y- 116
BSC1 Y- 5	3160 Y- 98	3279-3 Y- 216	3482 Y- 116
BSC2 Y- 8	3161 Y- 100	3279-3A Y- 217	3483 Y- 116
BSC3 Y- 11	3162 Y- 101	3279-3B Y- 223	3490 Y- 306

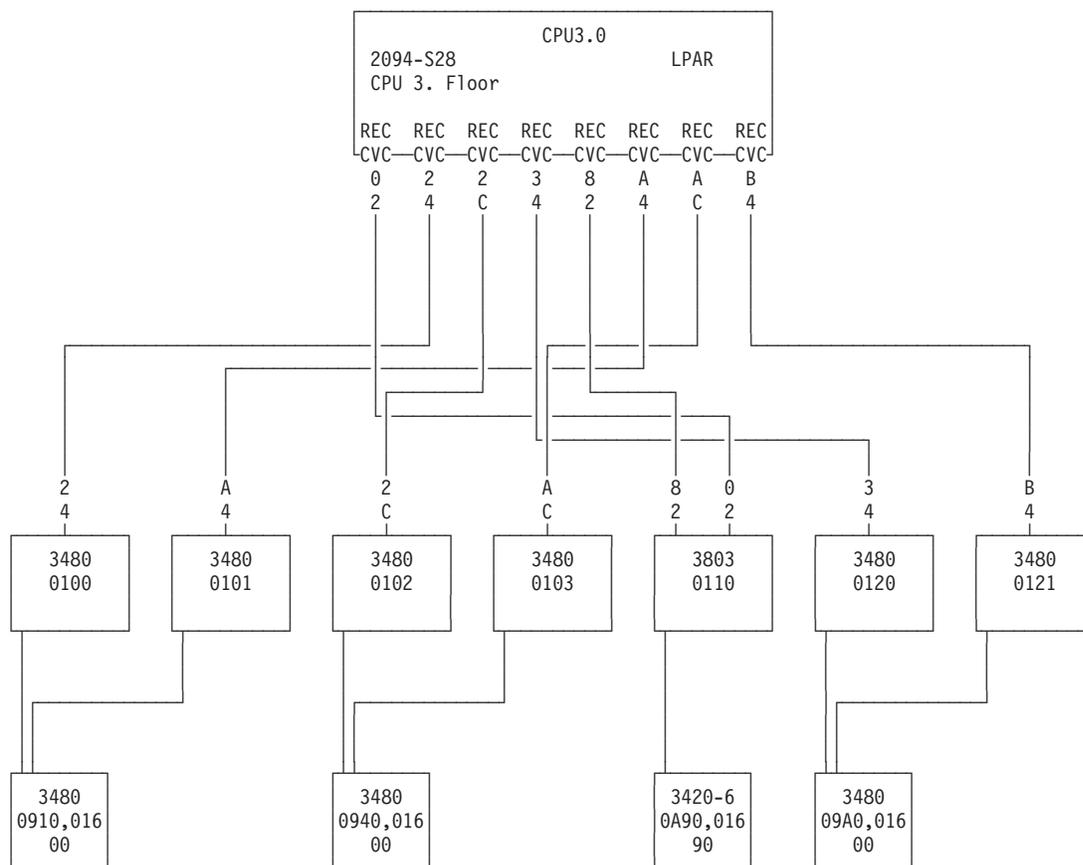
Figure 225. I/O Definition Reference (Part 5 of 5)

Graphical configuration reports

This section contains one example for an LCU report. “Create or view graphical configuration reports” on page 257 describes how to produce this report.

LCU Report

The LCU report shows all logical control units for the designated processor. Each diagram shows one or more logical control units.



IODF compare reports

The following figures show examples of IODF compare reports that can be produced by the IODF compare function of HCD. Not all possible reports are shown, and not all examples are shown completely. “HCD compare functions” on page 265 describes how to produce these reports.

For XMP processors, the processorID for the comparison is shown together with the channel subsystem in question, for example, XMP01.1

With all IODF reports where channel subsystems of XMP processors are involved, you can compare two channel subsystems.

You can also compare an SMP processor to a channel subsystem of an XMP processor. If this kind of comparison is limited by processor only, the SMP processor is compared to channel subsystem 0 of the XMP processor.

Processor Compare Report

Processor Compare Report	TIME: 10:40 DATE: 2012-10-21 PAGE A - 1
New IODF name: BOKA.IODF76.WORK.R111	Old IODF name: BOKA.IODF75.PROD.R111

PROC	New IODF	Old IODF	Description
GOLDENE1	Actual Data with 4 CSSes primary 2094-S38	Old Data with 4 CSSes	Processor Description continued
GOLDENE2	Added LPAR 2094-S28 with 4 CSSes secondary 2094-S28 80800002795A209409 + -11-2314:37:54 ERV01	80800002795A209409 + -11-1316:53:04	Processor Configuration Mode Processor Type Processor Serial Number Processor Network Name of SNA Address Processor CPC Name of SNA Address Processor Description continued Processor Token continued Processor Local System Name

Figure 226. Processor Compare Report

Channel Subsystem Compare Report

1

Channel Subsystem Compare Report

TIME: 13:03 DATE: 2014-06-23 PAGE A - 2

New IODF name: BVTH.IODFA1.DOCU.Z22.WORK

Old IODF name: BVTH.IODFA0.DOCU.Z22.WORK

Limited to New Processor Id: P2964

Old Processor Id: P2964

P2964	4	Actual Data	Old Data	
		CSS 4 of P2964	same	Channel Subsystem Description
		65280	same	Maximum Numbers of Devices in Subchannel Set 0
		65535	same	Maximum Numbers of Devices in Subchannel Set 1
		65535	same	Maximum Numbers of Devices in Subchannel Set 2
		65535	same	Maximum Numbers of Devices in Subchannel Set 3
P2964	5	Actual Data	Old Data	
		modify CSS 5 SS1		Channel Subsystem Description
		65280	same	Maximum Numbers of Devices in Subchannel Set 0
		65000	65535	Maximum Numbers of Devices in Subchannel Set 1
		65535	same	Maximum Numbers of Devices in Subchannel Set 2
		65535	same	Maximum Numbers of Devices in Subchannel Set 3

Figure 227. Channel Subsystem Compare Report

PCIe Function Compare Report

The *PCIe Function Compare Report* shows the changes in the IDs and attributes of PCIe functions between processors of two IODFs.

New IODF name: BVTH.IODFA0.DOCU.Z22.UID1.WORK

Old IODF name: BVTH.IODFA0.DOCU.Z22.UID.WORK

Limited to New Processor Id: P2964V3 Old Processor Id: P2964V3

PROC	FID	New IODF	Old IODF	Description
P2964V3	0013		Deleted	
			111 ROCE 3 F013 VF and 2 pnetids, + a=lp01 PNETID1 ID1302	Channel ID (CHID) Function Type Virtual Function ID (VF) UID parameter Function Description continued Physical Network ID 1 Physical Network ID 2 Physical Network ID 3 Physical Network ID 4
			>> LP01	Partition in Access List
			>> LP21 >> LP22 >> LP23 >> LP24	Partition in Candidate List Partition in Candidate List Partition in Candidate List Partition in Candidate List
P2964V3	0403		Deleted	
			7C4 ISM 4 undefined ism a=lp16 CONN2	Channel ID (CHID) Function Type Virtual Function ID (VF) UID parameter Function Description continued Physical Network ID 1 Physical Network ID 2 Physical Network ID 3 Physical Network ID 4
			>> LP11	Partition in Access List
P2964V3	0FF0	Actual Data	Old Data	
		undefined	0FF0	UID parameter

Figure 228. PCIe Function Compare Report

Partition Compare Report

Partition Compare Report

TIME: 10:22 DATE: 2016-04-27 PAGE B - 2

New IODF name: BVTH.IODFA0.DOCU.Z22.UID1.WORK

Old IODF name: BVTH.IODFA0.DOCU.Z22.UID.WORK

Limited to New Processor Id: P2964V3 Old Processor Id: P2964V3

PROC	PART	New IODF	Old IODF	Description
P2964V3.0	LP01	Actual Data	Old Data	
		1	same	Partition Number
		OS	same	Partition Usage
		N	Y	Partition UID flag
		Test partition 01	same	Partition Description

Figure 229. Partition Compare Report

Channel path compare report

Channel Path Compare Report

TIME: 12:46 DATE: 2015-11-26 PAGE C - 1

New IODF name: THI.IODFA0.WORK

Old IODF name: THI.IODFA0.COMP

Limited to New Processor Id: GA22964 Old Processor Id: GA22964

PROC	CHPID	New IODF	Old IODF	Description
GA22964.0	01	Actual Data	Old Data	
		7E2 IQD DED undefined not managed none 16 X IEDN	7E0 same same same same same same	Channel ID (CHID) Channel Path Type Channel Path Operation Mode Channel Path Description Connects to Dynamic Switch Channel Path Managed Indicator Channel Path I/O Cluster Name MTU size (in KB) IQD Function Physical Network ID 1
		>> OS01	>> same	Partition in Access List
GA22964.0	02	Added		
		400 OSD DED undefined not managed none not disabled PNETOSD1 PNETOSD2		Channel ID (CHID) Channel Path Type Channel Path Operation Mode Channel Path Description Connects to Dynamic Switch Channel Path Managed Indicator Channel Path I/O Cluster Name Queue Prioritization Physical Network ID 1 Physical Network ID 2 Physical Network ID 3 Physical Network ID 4
		>> OS01		Partition in Access List
GA22964.0	60	Actual Data	Old Data	
		CS5 SPAN undefined not managed none * 1	same same same same same same same	Channel Path Type Channel Path Operation Mode Channel Path Description Connects to Dynamic Switch Channel Path Managed Indicator Channel Path I/O Cluster Name HCA Adapter ID HCA Port
		>> OS01	>> same	Partition in Access List
		>> OS0F	>> same	Partition in Candidate List
		>> GA2#CS5.0 00 >> GA2#CS5.3 00	>> same >> same	Attached to Processor, Channel Path Attached to Processor, Channel Path
		>> FFFA	>> same	Attached Control Unit
		>> FF85,8	>> same	Attached Device, Range

Figure 230. Channel Path Compare Report

Control Unit Attachment Compare Report

Control Unit Attachment Compare Report

TIME: 14:21 DATE: 2012-10-21 PAGE D - 4

New IODF name: BOKA.IODF38

Old IODF name: BOKA.IODF38.TEMP

Limited to New Processor Id: FR38LPAR Old Processor Id: FR38LPAR

PROC	CU	New IODF	Old IODF	Description
FR38LPAR	2000	Actual Data	Old Data	
		serial	same	Control Unit Attachment Type
		S *	same	Protocol
		2 *	same	I/O Concurrency Level
		1	same	Control Unit Address
		00,256	same	Unit Address, Number of addresses
		>> 07.9B	>> same	Connected Channel Path, Destination Link Address
		>> 0C.70	>> same	Connected Channel Path, Destination Link Address
		>> 6B.9E	>> same	Connected Channel Path, Destination Link Address
		5	same	Number of Connected Managed Channel Paths
FR38LPAR	2200	Actual Data	Old Data	
		serial	same	Control Unit Attachment Type
		S *	same	Protocol
		2 *	same	I/O Concurrency Level
		3	same	Control Unit Address
		00,256	same	Unit Address, Number of addresses
		>> 0B.B9	>> same	Connected Channel Path, Destination Link Address
		>> 17.9B	>> same	Connected Channel Path, Destination Link Address
		>> 6B.9A	>> same	Connected Channel Path, Destination Link Address
		5	same	Number of Connected Managed Channel Paths

Figure 231. Control Unit Attachment Compare Report

Device Attachment Compare Report

Device Attachment Compare Report

TIME: 12:16 DATE: 2012-10-21 PAGE E - 11

New IODF name: BOKA.IODF75.WORK.R17

Old IODF name: BOKA.IODF77.WORK.R17

Limited to New Processor Id: GOLDENE1 Old Processor Id: GOLDENE1
 Limited to New CSS Id: 0 Old CSS Id: 0

PROC	Device, Range	New IODF	Old IODF	Description
GOLDENE1.0	0000,8	Actual Data	Old Data	
		0	1	Subchannel Set ID
		00	same	Unit Address
		Yes	same	Illegal Status Detection Facility
		No	same	Timeout Facility
		undefined	same	Preferred Channel Path
		>> GECSS0FX	>> same	Partition in Explicit Device Candidate list
		>> GECSS01X	>> same	Partition in Explicit Device Candidate list
		>> GECSS03X	>> same	Partition in Explicit Device Candidate list
GOLDENE1.0	0008,8	Actual Data	Old Data	
		0	same	Subchannel Set ID
		08	same	Unit Address
		Yes	same	Illegal Status Detection Facility
		No	same	Timeout Facility
		undefined	same	Preferred Channel Path

Figure 232. Device Attachment Compare Report

Control Unit Compare Report

Control Unit Compare Report

TIME: 08:41 DATE: 2012-10-21 PAGE F - 1

New IODF name: REDDE.IODF00.COMP1

Old IODF name: REDDE.IODF00.COMP2

Limited to New Processor Id: LPARPROC Old Processor Id: LPARPROC
 Limited to New Partition Id: PART1 Old Partition Id: PART3

CU	New IODF	Old IODF	Description
0200	Actual Data	Old Data	
	3990	same	Control Unit Type
		same	Control Unit Serial Number
		same	Control Unit Description
	>> BASPROC	>> new added	Attached to Processor
	>> LPARPROC	>> same	Attached to Processor
	>> P2084.1	>> same	Attached to Processor
	>> 0200,15	>> same	Attached Device, Range
	>> 020F	>> same	Attached Device, Range
	>> 0210,11	>> same	Attached Device, Range
	>> 01 E4	>> same	Connected to Switch, Port
	>> 01 E5	>> same	Connected to Switch, Port
	>> access list	>> candidate list	Relation to Limiting LPAR

Figure 233. Control Unit Compare Report

Device Compare Report

Device Compare Report

TIME: 08:41 DATE: 2012-10-21 PAGE 6 - 1

New IODF name: REDDE.IODF00.COMP1

Old IODF name: REDDE.IODF00.COMP2

Limited to New Processor Id: LPARPROC Old Processor Id: LPARPROC
 Limited to New Partition Id: PART1 Old Partition Id: PART3

Device, Range	New IODF	Old IODF	Description
0100	Actual Data	Old Data	
	9032	same same same same	Device Type Serial Number Device Description VOLSER
	>> BASPROC >> old deleted >> LPARPROC >> P2084.1	>> new added >> BASPROC1 >> same >> same	Attached to Processor Attached to Processor Attached to Processor Attached to Processor
	>> 0100	>> same	Connected to Control Unit
	>> access list	>> candidate list	Relation to Limiting LPAR
	0200,15	Actual Data	Old Data
3390 PROD4 AB - duplexed D		same same PROD4 AB same	Device Type Serial Number Device Description VOLSERn PPRC Usage
>> BASPROC >> LPARPROC		>> new added >> same	Attached to Processor Attached to Processor
>> 0200 >> 0210		>> same >> same	Connected to Control Unit Connected to Control Unit
>> OSOSOSOS		>> same	Attached to Operating System Configuration
>> access list		>> candidate list	Relation to Limiting LPAR

Figure 234. Device Compare Report

Switch Compare Report

Switch Compare Report

TIME: 13:52 DATE: 2012-10-21 PAGE H - 1

New IODF name: HCI.IODF00

Old IODF name: HCI.IODF01

SWITCH	New IODF	Old IODF	Description
01	Actual Data	Old Data	
	9032	same same same	Switch Type Switch Serial Number Description
	>> 1000 1000 >> 1010 1010 >> 1020 1020	>> same >> same >> same	Switch Control unit, switch device Switch Control unit, switch device Switch Control unit, switch device
	Actual Data	Old Data	
02	9032	same same same	Switch Type Switch Serial Number Description
	>> 2222 2222 >> 3333 3333 >> 4444 4444	>> same >> 4444 4444 >> new added	Switch Control unit, switch device Switch Control unit, switch device Switch Control unit, switch device
	Actual Data	Old Data	
	Actual Data	Old Data	

Figure 235. Switch Compare Report

Switch Detail Compare Report

Switch Detail Compare Report

TIME: 13:52 DATE: 2012-10-21 PAGE I - 9

New IODF name: HCI.IODF00

Old IODF name: HCI.IODF01

Limited to New Switch Id: 01 Old Switch Id: 01

SWITCH	PORT	New IODF	Old IODF	Description
01	B0	Actual Data	Old Data	
		installed	same	Port Installed Flag Port Name
		>> 0800 >> P2084.1 31	>> same >> same	Attached to Control Unit Attached to Processor, Channel Path
01	B1	Actual Data	Old Data	
		installed CU_400	same CU_500	Port Installed Flag Port Name
		>> 0800	>> 0500	Attached to Control Unit
01	B2	Actual Data	Old Data	
		installed PROCA_CP18 occupied	same not occupied	Port Installed Flag Port Name Port Occupied Flag
		Actual Data	Old Data	

Figure 236. Switch Detail Compare Report

Switch Configuration Compare Report

The Switch Configuration Compare Report compares the switch configurations contained in the specified IODFs.

Switch Configuration Compare Report TIME: 13:52 DATE: 2012-10-21 PAGE J - 1
 New IODF name: HCI.IODF00 Old IODF name: HCI.IODF01

SWITCH	SWCONFIG	New IODF	Old IODF	Description
01	BASIC	Actual Data	Old Data	
		PROHIBIT SW Building 01-125	ALLOW same	Default Connection Type Switch Configuration Description

Figure 237. Switch Configuration Compare Report

Switch Configuration Detail Compare Report

Switch Configuration Detail Compare Report TIME: 13:52 DATE: 2012-10-21 PAGE K - 1
 New IODF name: HCI.IODF00 Old IODF name: HCI.IODF01

SWITCH	SWCONFIG	PORT	New IODF	Old IODF	Description
01	BASIC	E1	Actual Data	Old Data	
			Unblocked	same	Blocked / Unblocked Connection
			>> F1	>> same	Port of Dedicated Connection
01	BASIC	E2	Actual Data	Old Data	
			Unblocked	same	Blocked / Unblocked Connection
			>> E7	>> same	Port of Allowed Connection
			>> F3	>> same	Port of Allowed Connection
			>> E8	>> same	Port of Prohibited Connection
01	BASIC	E3	Actual Data	Old Data	
			Unblocked	same	Blocked / Unblocked Connection
			>> E8	>> same	Port of Prohibited Connection
			>> E9	>> same	Port of Prohibited Connection

Figure 238. Switch Configuration Detail Compare Report

Esoteric Compare Report

Esoteric Compare Report

TIME: 08:32 DATE: 2012-10-21 PAGE P - 1

New IODF name: REDDE.IODF00.COMP1

Old IODF name: REDDE.IODF00.COMP2

Limited to New Operating System Id: OS1

Old Operating System Id: OS1

OSCONFIG EDT	ESOTERIC	New IODF	Old IODF	Description
OS1	01	BOBO	Deleted	
			No	Esoteric is VIO Eligible Esoteric Token
			>> 0201,3 >> 0205,3	Assigned Device, Range Assigned Device, Range
OS1	01	HUGO	Added	
			No	Esoteric is VIO Eligible Esoteric Token
			>> 0200,8 >> 0300,64	Assigned Device, Range Assigned Device, Range
OS1	01	SYSDA	Actual Data	
			Yes	Esoteric is VIO Eligible Esoteric Token
			>> old deleted >> 0300,64	Assigned Device, Range Assigned Device, Range
			Old Data	
			same same	Esoteric is VIO Eligible Esoteric Token
			>> 0204 >> same	Assigned Device, Range Assigned Device, Range

Figure 239. Esoteric Compare Report

Operating System Compare Report

Operating System Compare Report

New IODF name: BVTH.IODF01.PPRC

Old IODF name: BVTH.IODF01

OSCONFIG	New IODF	Old IODF	Description
CDRZOS2	Actual Data	Old Data	
	MVS copied D/R configuration DRZOS2	same same	Operating System Type Operating System Description continued
		same same	D/R Site Operating System ID Generated Operating System
DRZOS1	Added		
	MVS generated from ZOS + 1		Operating System Type Operating System Description continued
	Yes		D/R Site Operating System ID Generated Operating System
DRZOS2	Actual Data	Old Data	
	MVS generated from ZOS + 2	same same	Operating System Type Operating System Description continued
	Yes	same	D/R Site Operating System ID Generated Operating System

Figure 240. Operating System Compare Report

OS Device Compare Report

OS Device Compare Report

TIME: 08:32 DATE: 2012-10-21 PAGE R - 1

New IODF name: REDDE.IODF00.COMP1

Old IODF name: REDDE.IODF00.COMP2

Limited to New Operating System Id: OS1

Old Operating System Id: OS1

OS	Device, Range	New IODF	Old IODF	Description
OS1	0100	Actual Data 9033 SWCH Yes Yes	Old Data same same same same	Device Type Name of Generic Value(s) of Parameter OFFLINE Value(s) of Parameter DYNAMIC
OS1	0200,32	Actual Data 3390 3390 No * Yes	Old Data same same same same	Device Type Name of Generic Value(s) of Parameter OFFLINE Value(s) of Parameter DYNAMIC
OS1	01D1,4	Actual Data 3390A 3390 0 No Yes ... SHARED ...	Old Data same same 1 same same ... same ...	Device Type Name of Generic Subchannel Set ID Value(s) of Parameter OFFLINE Value(s) of Parameter DYNAMIC ... Feature ...

* indicates this value as default value (only shown when both sides exist)

Figure 241. OS Device Compare Report

Note: Device ranges in a subchannel set with a subchannel set ID > 0 are displayed in a 5-digit notation with the leading digit indicating the subchannel set ID. For example, a device range 1000,64 located in subchannel set 1 is shown as 11000,64. A device range 2000,32 in subchannel set 0 is shown as 2000,32.

OS Console Compare Report

OS Console Compare Report

TIME: 12:41 DATE: 2012-10-21 PAGE Q - 1

New IODF name: USER.IODF03.WORK

Old IODF name: HCD.IODF01.WORK

Limited to New Operating System Id: OPSYS01

Old Operating System Id: OPSYS01

OSCONFIG DEVICE	New IODF	Old IODF	Description
OPSYS01 0001	Actual Data 2	Old Data 1	Order Number
OPSYS01 0002	Added 1		Order Number

Figure 242. OS Console Compare Report

CSS / OS Device Compare Report

CSS / OS Device Compare

TIME: 17:00 DATE: 2012-10-21 PAGE S - 1

IODF Name : SEL.IODF00.WORK1
 Processor Id: PROC01 Cms Id: 0 Partition Name: PART00 ./ Operating System Configuration Id: MVS1

Device, Range		CSS Device Type	OS Device Type
A000,80	*	3390B	same
A050,176	-	3390A	same
A200,80	-	3390B	same
E210,16	-	3390	3380
E220,1000		3390	same
1017F,128	-		3390A
10220,64	-		3390A
10310,48	-		3390A
2F900	-	3390S	same
2F901	-	3390D	same
2F902	-	3390S	same

* Devices relate to the limiting LPAR via CHPIDs which have the limiting LPAR in the candidate list only.
 - Devices relate to the limiting LPAR via CHPIDs but the device is excluded from the CSS with an explicit device candidate list.

Figure 243. CSS / OS Device Compare

Note: A device range in a subchannel set with a subchannel set ID > 0 is displayed in a 5-digit notation, with the leading digit indicating the subchannel set ID. For example, 10310,48 denotes the devices 0310 with range 48 in subchannel set 1.

Appendix C. Problem determination for HCD

Overview

The information in this appendix is intended to help you diagnose problems that may arise with definitions that were created by the use of HCD. It explains:

- How to identify problems
- What diagnostic information and tools you can use
- How to search problem reporting data bases

Because HCD is part of z/OS, problems with HCD must generally be handled as explained in *z/OS V2R2 Problem Management*.

Product Identifiers

- Module Prefix: CBD
- Component ID: 5695SC1XL

Identifying problems

Before you can begin to diagnose a system problem, you have to know what kind of problem you have.

The following table contains examples of symptoms you can use to determine a problem. Each symptom refers to a corresponding section for further problem diagnostic.

Table 30. Symptoms of system problems

Symptom	Corresponding Section
HCD terminates abnormally	"HCD abnormal termination" on page 470.
Wait State during IPL	"Error during IPL (Wait State Codes)" on page 472.
A function key assignment does not match the functions that can be performed on the panel.	"Problems with panels and function key assignment" on page 473
Messages CBDA400I to CBDA420I are displayed	"Problems with help information provided by HCD" on page 474
Output of textual report is incorrect or incomplete	"Problems with output of HCD textual reports" on page 475
Output of graphical report is incorrect or incomplete	"Problems with output of HCD graphical reports" on page 475
Messages during initialization of HCD	"Problems during initialization of HCD" on page 477
A string like ?PARMnn? appears on the Define Device Parameters / Features panel	"Problems with UIMs" on page 478
A UIM is flagged in error on the Installed UIMs panel	"Problems with UIMs" on page 478
Messages during migration	"Resolving migration errors" on page 322
HCD does not display an error message when you make a mistake	"HCD internal problems" on page 479

Table 30. Symptoms of system problems (continued)

Symptom	Corresponding Section
An HCD generated IOCP input data set fails when using the IOCP program	"HCD internal problems" on page 479
<i>Transmit configuration package</i> action does not produce the expected results	"Problems with the Transmit Configuration Package action" on page 480
HCD LDAP backend terminates abnormally	"HCD LDAP backend abnormal termination" on page 485
HCD LDAP backend returns one of the error codes LDAP_OPERATIONS_ERROR, LDAP_NO_MEMORY, or LDAP_LOCAL_ERROR	"Adverse LDAP return code from the HCD LDAP Backend" on page 486
Message CBD0009E is printed to the started task's log	"Error in the HCD LDAP backend" on page 486

HCD abnormal termination

If HCD terminates abnormally, view the HCD message log that contains the termination message CBDA000I specifying the *system abend code* (also called *abend code*) and the *reason code* in the HCD message log.

Refer to the message descriptions shown in *z/OS and z/VM HCD Messages*. Take the action as described under "Programmer Response". If the message points to a probable logic error in one of the modules of HCD, develop a search argument for the problem-reporting data bases. If the search finds that the problem has been reported before, request the problem fix; if not, report the problem to IBM. For a list of additional information that should be provided, see the appropriate message explanation.

Diagnosing system abend code '00F'

If HCD terminates with system abend code '00F', this abend code is accompanied by a reason code, which refers to one of the HCD messages describing the reason of the failure. Note that you have to view the HCD message log for the system abend code and reason code. The reason code consists of eight digits and has the format 'mnnnllll' where:

m Is the prefix indicating the HCD message range:

- 0** CBDAxxxx messages
- 1** CBDBxxxx messages
- 2** CBDCxxxx messages
- 6** CBDGxxxx messages

nnn Is the message number within the HCD message range.

llll Is the message reason code describing in more detail the reason of the message.

The information provided by the abend code can be used as a quick reference into the message. For example, the reason code 00990106 means that:

- The message CBDA099I was issued.
- The message reason code is 0106.

The reason code 00150095 means that:

- The message CBDA015 was issued.

- The message reason code is 95.

Table 31 and Table 32 show what the search argument and the problem data could look like.

Table 31. Search Argument

Search Argument	Description	Example
AB/S0hhh	System abend code	AB/S000F
PRCS/mnnnlll	Reason code	PRCS/00990106
MS/ccccnns	Message identifier	MS/CBDA099I

Table 32. Problem Data

Problem Data	Example
CBDA000 abend code and reason code	00F and 00990406
CBDA099 reason code	406
CBDA099 additional error information (content of HCDMLOG). It is important that <i>all</i> information shown in HCDMLOG is recorded.	
The ID of the panel where the error occurred	CBDPPRF0
Description of what type of action the user wanted to perform when the problem occurred	Add a Processor
The TRACE output data set (See "TRACE command" on page 491 for instructions how to produce an HCD trace output.)	

ISPF list file and abend panel: Additional information on errors may be recorded in the ISPF list file.

For abends, additional information may be displayed on the ISPF abend panel.

Diagnosing system abend other than '00F'

If HCD terminates with an abend code other than '00F' (indicated in the terminating message), proceed as follows:

1. Look at the explanation of the abend code and any reason code that accompanies the abend code. Take the recommended actions.
2. Look for any messages that accompany the abend. Take the recommended actions.
3. Obtain the SYS1.LOGREC record. (Format the SYS1.LOGREC record using EREP.)
4. In SYS1.LOGREC find the SDWAVRA information which is as follows:
 - The CSECT (module) names found in the diagnostic stack.
The CSECT names are separated by a blank. The SDWAVRA contains all CSECT names from the diagnostic stack as long as they fit into it. If the SDWAVRA is too small to contain all names, the premature end of the CSECT name trace is indicated by an asterisk.
 - The data from each diagnostic stack entry that is marked as VRA data.

This is normally the input parameter list of the modules corresponding to the CSECT name trace.

Table 33 and Table 34 show what the search argument and the problem data associated with our example could look like.

Table 33. Search Argument

Search Argument	Description	Example
RIDS/CBDcccc	CSECT name	RIDS/CBDMGHCP
AB/S0hhh	System abend code	AB/S0106
PRCS/mnnnllll	Reason code	PRCS/0000000B
MS/cccnnns	Message identifier	MS/CSV011I
FLDS/SDWAVRA VALU/cccc	SDWAVRA contents	

Table 34. Problem Data

Problem Data	Example
SYS1.LOGREC error record	
SDWAVRA information	
Accompanying messages	
Component ID and FMID	
Linkage editor output	
Description of what type of action the user wanted to perform when the problem occurred	
The TRACE output data set (See "TRACE command" on page 491 for instructions how to produce an HCD trace output.)	

Error during IPL (Wait State Codes)

IOS may issue wait state codes during IPL when using an IODF to perform an IPL. The wait state codes indicate that there is a problem, for example, with an IODF data set or with device specifications in the IODF or UIM. The reason codes with the wait state codes point to the cause of the problem. For information about the codes, refer to *z/OS MVS System Codes*.

Depending on the code that was issued, you have to use the arguments that apply to the specific situation listed in the box Table 35. The same applies for submitting problem data.

Table 35 and Table 36 on page 473 show what the search argument and the problem data could look like.

Table 35. Search Argument

Search Argument	Description	Example
WS/D0hhh	Wait state code	WS/D0083
PRCS/mnnnllll	Reason code	PRCS/00000002

Table 35. Search Argument (continued)

Search Argument	Description	Example
MS/cccnns	Message identifier	
PIDS/name of UIM	Program name	PIDS/CBDUS005
VALU/Cccccccc (if applicable)	Message variable text	

Table 36. Problem Data

Problem Data	Example
Wait State Code	D0083
Reason Code	01
Accompanying message	
UIM name (if available)	CBDUS005
Stand-alone dump	
IODF dump	

Problems with panels and function key assignment

If problems with panels or the assignment of function keys occur, ensure the following:

- Data set SYS1.SCBDTENU must be allocated to ISPTLIB and data set SYS1.SCBDPENU to ISPLIB.
- SYS1.SCBDCLST must be allocated to SYSPROC
- SYS1.SCBDPENU, SYS1.SCBDMENU, and SYS1.SCBDTENU must be dynamically allocated when HCD has been started. Compare with LIBDEF definitions in CBDCHCD.

If the library allocation is correct, develop a search argument, and if no problem solution is found, report the problem. To display the panel identifier, use the ISPF command PANELID. The name of the function panel will be shown in the upper left corner of the panel.

Table 37 and Table 38 show what the search argument and the problem data could look like.

Table 37. Search Argument

Search Argument	Description	Example
RIDS/CBDcccc	Panel identifier	RIDS/CBDPHW10

Table 38. Problem Data

Problem Data	Example
Panel identifier and name of the panel where the error was detected.	CBDPHW10 Define, Modify, or View Configuration Data
Type of error found.	

Problems with help information provided by HCD

Messages that relate to problems with the HCD help facility have the identifiers CBDA400I to CBDA420I. Use the commands:

- HELPID to display the name of the help panel at the end of the command line. The name is displayed in the command line just before the scroll field. It can be used as search argument.
- HELPTEST to display the help panel while in help mode. It allows you to review or test any help panel while in help mode. That is, it eliminates the need to create the appropriate situation if a review or test of a help panel is required. This command can also be used to get the help information for a message. The help member for a message consists of the message ID minus the trailing severity indicator (such as "I"). For example, the help member for message CBDA200I has the name CBDA200. Thus you may get an explanation for messages that are not yet listed in the messages documentation.

Problem with content, wording, mismatch

If problems with content, wording, or mismatches are encountered, obtain the help panel name by using the HELPID command. The help panel name is displayed at the end of the command line.

Table 39 shows what the search argument could look like.

Table 39. Search Argument

Search Argument	Description	Example
RIDS/CBDcccc	Help panel name	RIDS/CBDF403
RIDS/CBDcccc	Panel identifier	RIDS/CBDPDPVF0

Problem shown by help messages CBDA400I or CBDA405I

If message CBDA400I or CBDA405I is issued, check:

1. The library concatenation for your HCD invocation.
2. That the help members are installed in the proper libraries (in SYS1.SCBDHENU).
3. That the help library is allocated either in LINKLST member or to ISPLLIB.

Problem shown by help messages other than CBDA400I and CBDA405I

If a message in the range CBDA400I to CBDA420I but other than CBDA400I and CBDA405I occurs, you have probably encountered a logic error in the dialog.

Table 40 and Table 41 on page 475 show what the search argument and the problem data could look like.

Table 40. Search Argument

Search Argument	Description	Example
MS/cccnns	Message identifier	MS/CBDA404
RIDS/CBDcccc	Help panel name	RIDS/CBDF401

Table 41. Problem Data

Problem Data	Example
Message identifier of the message that was issued. All additional information shown in the message, such as the name of the help panel or the reference phrase.	CBDA404
Panel identifier	CBDPDVF0
Help panel name	CBDF401
Description of the related field (in case of field help).	Action entry field
Name of the action choice (in case of action bar help).	
Name of the command (in case of command help).	

Problems with output of HCD textual reports

Use this procedure if you find problems that relate to the HCD report facility.

1. Check the output of the report job for messages that provide additional information.
2. For incorrect or incomplete output:
 - a. Compare the contents of the IODF with the output. For example, if the device features are not shown correctly in the OS device detail report, use the action *View device definition* on the I/O Device List to display the definitions of the device for which the report seems to be incorrect.
 - b. If you find out that the definitions in the IODF are correct, but the report output is incorrect, report this problem to IBM.

Table 42 shows what the problem data could look like.

Table 42. Problem Data

Problem Data	Example
Type of report that was to be created.	Control unit detail report
JCL that was used to create the report.	
The content of the IODF for which the report was requested. Refer to "TRACE command" on page 491 to create an IODF dump.	

Problems with output of HCD graphical reports

When you have any problems printing or viewing a graphical configuration report, read the following problem descriptions. If you have any other problem, report the problem to IBM.

Screen shows four dots

When displaying the configuration, the screen shows only four dots.

Check that the background and foreground color you specified in the HCD profile match. Choose a foreground color that is visible on the background color.

Incorrect DCF, GDF, or GML format in output data set

You can simply check, whether you have created the output format you have specified in the profile using the keyword GCR_FORMAT. Browse or edit the created data set.

BookMaster format looks like this:

```
:userdoc
.layout 1
.dr thick weight .4mm
.rh on
.sp 2
```

DCF format looks like this:

```
.df graph font X0GT20
.tr 31 AC BE BC 76 AB 30 BB 15 8F 77 CB 78 CC 80 EB 64 EC 6A FA 24 BF
.ll 240mm
.dr thick weight .4mm
.rh on
.sp 2
```

GML format looks like this:

```
:gdoc
.df graph font X0GT20
.tr 31 AC BE BC 76 AB 30 BB 15 8F 77 CB 78 CC 80 EB 64 EC 6A FA 24 BF
.ll 240mm
.dr thick weight .4mm
.rh on
.sp 2
```

If the output is not correct, make sure that:

- The HCD profile is allocated with ddname HCDPROF before invoking HCD.
- The profile contains the keywords GCR_FORMAT=DCF or GML and GCR_FONT with an appropriate font (for example, X0GT20 for 3820 printers).

For information on how to specify keywords in the HCD profile, refer to “Defining an HCD profile” on page 19.

Illegible printout when using DCF or GML

The output of a report data set contains correct DCF and GML format, but the printout is illegible.

Make sure that:

- A monospaced font (for example, X0GT20 for 3820 printers) is specified in the HCD profile using the keyword GCR_FONT.
- The specified font is installed on your printer.

Output exceeds page boundary

The printed output exceeds page boundary.

Make sure that:

- During printing you specified a parameter to print the report in landscape format, that is to rotate the printout by 90 degree.
- The parameter LAYOUT 1 was specified to use the full page for the report.

Box characters are not correct

When using DCF formatting, the box characters are not correct.

The graphical print facility uses special hex characters for the various box characters. These special characters are then translated to real box characters by means of the .tr command. If the selected font does not contain the box characters, you must either choose another font or modify the .tr command in the file generated by the graphical print facility. See the following table for information on which hex combinations HCD uses for the various box characters.

```
.tr 31 AC BE BC 76 AB 30 BB 15 8F 77 CB 78 CC 80 EB 64 EC 6A FA 24 BF
```

where

31	Upper left corner	78	Downward T
BE	Upper right corner	80	Leftward T
76	Lower left corner	64	Rightward T
30	Upper right corner	6A	Bar
15	Junction (+)	24	Hyphen (dash)
77	Upward T		

Table 43 shows what the problem data could look like.

Table 43. Problem Data

Problem Data	Example
Type of report that was to be created.	LCU report
The content of the IODF for which the report was requested. Refer to "TRACE command" on page 491 to create an IODF dump.	

Problems during initialization of HCD

If a problem occurs during initialization, HCD does one of the following:

- Issues a message and continues the initialization
- Terminates the initialization

Whether the initialization of HCD *continues* or *terminates* depends on the error that is encountered as explained in the following.

Initialization continues

If a UIM service routine encounters an error during initialization, HCD works without this UIM. HCD pops up messages on the user's terminal to inform the user that messages were written to the message log.

Note: If an error is encountered in a UIM *and* if SYSUDUMP is allocated, HCD does not continue. An HCD abend '00F' is forced to provide a dump at the point where the error was detected.

Use the option *List Installed UIMs* to display the panel "Installed UIMs". On this panel, the UIM is marked as in error. (Refer to "Query installed UIMs" on page 276.)

If you do not have access to any UIM, check if your UIMs are correctly installed. In the HCD profile you can specify the name and volume serial number of the

library that contains the UIMs (see “Defining an HCD profile” on page 19). If you do not specify a name in the profile, SYS1.NUCLEUS is assumed as default names for the UIMs.

Initialization is terminated

The initialization is terminated either with an abend or with a message.

In case of an abend, the dialog:

- Pops up message CBDA040I on the user's terminal. The message informs the user that HCD has abnormally terminated.
- Puts message CBDA050I with abend code '00F' in the message log. The message also provides a reason code.
- Puts the message that is issued by a UIM service routine in the message log.

If an error is encountered in a UIM *and* if SYSUDUMP is allocated, an HCD abend '00F' is forced to provide a dump at the point where the error was detected.

Message CBDA041I: Means that HCD is not able to find the UIMs. If this message is issued during initialization using the "CIT" variable, make sure that the UIMs are installed in SYS1.NUCLEUS.

Table 44 and Table 45 show what the search argument and the problem data could look like.

Table 44. Search Argument

Search Argument	Description	Example
MS/cccnns	Message identifier	MS/CBDA041I
PIDS/UIM name	UIM name	

Table 45. Problem Data

Problem Data	Example
UIM name	
Message ID(s) and full message text	CBDA041I
TRACE output data set	
Refer to “TRACE command” on page 491.	

Problems with UIMs

For information on converting and testing UIMs, refer to *z/OS MVS Device Validation Support*.

The following explanations apply to UIMs provided by the installation and to UIMs provided by IBM.

Messages during initialization of HCD

Internal logic errors in UIMs are primarily found during the initialization of HCD. Refer to “Problems during initialization of HCD” on page 477 for information on how to proceed in case of initialization problems.

UIM problems after initialization of HCD

Internal logic errors in UIMs may also be discovered:

- During the definition of a device (as a string like ?PARMnn? on the Device Parameter Feature panel). Message CBDA381I indicates that you may have installed a back-level UIM.
- On the Installed UIMs panel when a UIM is flagged in error. In this case messages CBDA070I or CBDA096I may be issued. Message CBDA070I means that the UIM does not match the corresponding UDT. Message CBDA096I means an unresolvable conflict between a VM and MVS UIM.

You can use the message log, together with SYSUDUMP and HCDTRACE to find the error in the UIM that failed.

If the error relates to an installation-provided UIM, make appropriate corrections. For information on converting UIMs, refer to *z/OS MVS Device Validation Support*.

If the error relates to a UIM provided by IBM, report the problem.

Table 46 and Table 47 show what the search argument and the problem data could look like.

Table 46. Search Argument

Search Argument	Description	Example
MS/cccnns	Message identifier	CBDA070I
PIDS/UIM name	UIM name	CBDUS025

Table 47. Problem Data

Problem Data	Example
UIM name	CBDUS025
Message ID(s) and full message text	CBDA070I
Type of action the user wanted to perform	List installed UIMs
TRACE output data set	
Refer to "TRACE command" on page 491.	

HCD internal problems

When you have one of the following error situations, you probably have detected an internal HCD error:

- HCD displays wrong messages or does not display a message at all when you made a mistake.
- An HCD generated IOCP input data set causes error messages when using the IOCP program.

Report problems like this to IBM.

Table 48 and Table 49 on page 480 show what the search argument and the problem data could look like.

Table 48. Search Argument

Search Argument	Description	Example
MS/cccnns	Message identifier	
RIDS/CBDcccc	Panel identifier	RIDS/CBDPCH30

Table 49. Problem Data

Problem Data	Example
Message ID(s)	
Panel identifier	CBDPCH30
Type of action the user wanted to perform	Change Channel path definition
Description of configuration	

Problems with the Transmit Configuration Package action

If a problem occurs during the *Transmit configuration package* action, HCD may:

- Not start the action due to authorization problems
- Not submit the job
- Submit the job but not complete it

Job steps of the Transmit Procedure

Transmit configuration package builds a batch job with multiple steps. The step names are:

GO Creates an IDCAMS CLUSTER for a temporary work IODF.

ALLOCT2

Creates an IDCAMS CLUSTER for a temporary production IODF.

INIT1 Initializes the temporary work IODF.

INIT2 Initializes the temporary production IODF.

BLDPR1

The processor configurations that are contained in the configuration package are built into a data set as control statements. The processor list is specified via DD name HCDCNTL. If a processor contains a CFS channel path that has a connection to a CF partition external to the configuration package, the processor containing the CF partition is also included in the output data set.

MIGRPR1

The generated processor configuration control statements are migrated into the temporary work IODF while preserving the processor tokens from the master IODF.

PRINTPR1

The MESSAGES and LISTING data set are deleted if no error occurred.

BLDOS1

All OS configurations included in the configuration package are built into a data set as control statements. The OS configuration list is specified via DD name HCDCNTL.

MIGROS1

The generated OS configuration control statements are migrated into the temporary work IODF.

PRINTOS1

The MESSAGES and LISTING data set are deleted if no error occurred.

BLDSW1

The switch configurations of all switches containing ports that are connected to either a channel path or control unit of the processors of the

configuration package are built into a data set as control statements. The switch list is specified via DD name HCDCNTL.

MIGRSW1

The generated switch configuration control statements are migrated into the temporary work IODF.

PRINTSW1

The MESSAGES and LISTING data sets are deleted if no error occurred.

BPROD

A temporary production IODF is built from the temporary work IODF.

EXPOATT

The temporary production IODF is exported attended to the specified user/node ID.

EXPOUATT

The production IODF is exported unattended to the specified system.

DEL1 The temporary work IODF is deleted.

DEL2 The temporary production IODF is deleted.

Note: Stepname GO is used for HCDDECK, HCDMLOG, HCDLIB, HCDTRACE, and HCDPROF. Thus, the data sets are made available to the steps which require them.

Temporary data sets created by the Transmit Procedure

The transmit procedure creates the following data sets:

- hlq.IODFxx.zzzz (production IODF)
- hlq.IODFxx.XMIT.package.WORK (work IODF)
- hlq.IODFxx.XMIT.package.DECK (configuration decks)
- hlq.IODFxx.XMIT.package.MSGLOG (HCDMLOG)
- hlq.IODFxx.XMIT.package.sss.MESSAGES (HCDPRINT migration messages)
- hlq.IODFxx.XMIT.package.sss.LISTING (HCDASMP migration listing)

where:

hlq	is the high level qualifier specified on the transmit panel or the HLQ parameter of the batch utility.
xx	is the suffix of the target IODF name specified with the package
package	is the name of the configuration package to be transmitted
zzzz	are the qualifiers 3-n of the target IODF name
sss	qualify the migration type (PR1 for processor, OS1 for operating systems, SW1 for switch configurations)

After a successfully completed transmit action all these data sets, except the message log file, are deleted. The message log file is preserved until it is overwritten, when another transmit action using the same package name and IODF suffix is performed.

Apart from the production IODF, all redundant data sets remaining from a cancelled transmit action are identified by their common data set name qualifiers.

hlq.IODFxx.XMIT.package.

Authorization problems

Because the last sent date of the IODF from which the transmit action is performed is updated with the current date, you require write access to the accessed production IODF. Otherwise message CBDG247I is displayed.

You also need permission to write to the data sets with the qualifiers of the IODF to be created and transmitted.

Job is not submitted

If the work IODF or production IODF to be created temporarily exists already, it is not possible to start the transmit action. This may happen when a previous transmit job was cancelled, ended with an error or another transmit job is running which uses the same high level qualifier and target IODF name.

Depending on the source of the problem this may be resolved by deleting the existing temporary IODFs or by specifying a different high level qualifier for the target IODF.

Job is not completed

The transmit action generates a batch job. Check the HCD message log file to find out if the job was executed. It is shown as a sequence of HCD batch job steps ending with a successful export message. There are several possible causes if this is not the case:

- JCL errors: check the job output. To see all statements including the inline statements which are generated by the HCD dialog and submitted, issue

```
TRACE ON ID=JCL
```

and perform the transmit action. For an example of a trace refer to “Customization unsuccessful.”

- If the HCD message log file shows that a particular step failed, check the job output for potential allocation problems.
- If one of the migration steps failed, check the LISTING and MESSAGES data sets. (Refer to “Temporary data sets created by the Transmit Procedure” on page 481 for more information on the data sets created during the action.) Ensure that the same versions of UIMs are available for the *Transmit configuration package* action as for creation of the IODF.

The MESSAGES and LISTING data sets, as well as the generated decks, are deleted if no errors occurred. If you want to keep them, you can modify the conditional statements in procedure CBDJXMIT for the migrate steps. Do this by copying CBDJXMIT to a new procedure.

Proceed as follows, if you need to trace particular steps:

1. Define a profile including a TRACE statement.
2. Specify the stepname.HCDPROF DD with the profile name.
3. Allocate a trace data set name.
4. Specify GO.HCDTRACE DD with the name of the trace data set in order to use it for all steps to be traced or stepname.HCDTRACE to use the trace data set only for the single step.
5. To specify HCDDECK, HCDMLOG, HCDLIB, HCDTRACE, or HCDPROF use GO as the step name. The other steps refer to the definitions in the GO step. If you want to preserve a specific output data set, pre-allocate it to HCDDECK (see “Build I/O configuration data” on page 339).

Customization unsuccessful

This section describes points to be considered when customizing the transmit procedure.

The transmit procedure exploits the migration batch utility, which uses parsing macro CBDZPARS (residing in SYS1.MACLIB). If you want to use a different macro library, specify this as GO.HCDLIB

The dialog always generates and submits the following statements:

- All parameters for procedure CBDJXMIT
- The JOB card, JOBLIB and overwrite statements given by the user
- IDCAMS DEFINE CLUSTER and DELETE CLUSTER statements in steps GO, ALLOCT2, DEL1 and DEL2
- An HCDCNTL DD statement for at least one of BLDPR1, BLDOS1 or BLDSW1 job steps, dependent on the package content
- The SYSTSIN for EXPOATT or EXPOUATT, depending on whether attended or unattended export is selected

The following is a sample trace showing the batch job built by a transmit action.

```
09:53:01 97-11-04 Trace started.
//XMIT JOB (3243),'OS390H1',MSGCLASS=X,CLASS=A,REGION=4M
//JOBLIB DD DSN=SYS1.SCBDHENU,DISP=SHR
//XMT0 EXEC PROC=CBDJXMIT,PR=1,OS=1,SW=1,
// ATTEND=1,RECORDS='1684',
// DESC1='IODFST',DESC2='IODF88',
// QUALX='IODF88.XMIT.CB88',
// IODFTP='OS390H1.IODF88',
// IODFTW='OS390H1.IODF88.XMIT.CB88.WORK',
// IODFSP='IODFST.IODF11.MASTER'
//GO.SYSIN DD *
DEFINE CLUSTER(
NAME ( OS390H1.IODF88.XMIT.CB88.WORK.CLUSTER) -
LINEAR
RECORDS (1684) -
VOLUMES (DS7001)) -
DATA(NAME(OS390H1.IODF88.XMIT.CB88.WORK))
/*
//ALLOCT2.SYSIN DD *
DEFINE CLUSTER(
NAME ( OS390H1.IODF88.CLUSTER) -
LINEAR
RECORDS (1684) -
VOLUMES (DS7001)) -
DATA(NAME(OS390H1.IODF88))
/*
//BLDPR1.HCDCNTL DD *
CB88
CF14 ,CF
/*
//BLDOS1.HCDCNTL DD *
B710
/*
//BLDSW1.HCDCNTL DD *
71
72
74
77
/*
//EXPOATT.SYSTSIN DD *
CALL 'SYS1.LINKLIB(CBDMGHCP)', +
'EXPORT,OS390H1,PKSTCB88'
/*
//DEL1.SYSIN DD *
DELETE OS390H1.IODF88.XMIT.CB88.WORK.CLUSTER
/*
```

```
//DEL2.SYSIN DD *
      DELETE OS390H1.IODF88.CLUSTER
/*
09:53:29 97-11-04 Trace stopped.
```

Table 50 shows what the problem data could look like.

Table 50. Problem Data

Problem Data	Example
Job output	See example trace shown in section "Customization unsuccessful" on page 482.
Message log file	
Trace of failing step	
Deck for failing step	
LISTING data set	See "Job steps of the Transmit Procedure" on page 480
MESSAGES data set	See "Temporary data sets created by the Transmit Procedure" on page 481
Submitted job (via TRACE ID=JCL)	
Procedure used (if modified)	

LDAP problem determination

In addition to the regular text messages from the HCD LDAP backend, debug output can be requested. This debug output is switched on or off by specifying a debug level for the IBM Tivoli Directory Server for z/OS (see *z/OS IBM Tivoli Directory Server Administration and Use for z/OS*) and is printed to the same location as the HCD LDAP backend text messages. HCD LDAP backend's debug output is only available in English language and character representation IBM-1047.

The HCD LDAP backend mainly supports two debug levels: LDAP_DEBUG_ERROR and LDAP_DEBUG_TRACE.

LDAP_DEBUG_ERROR causes all the information concerning errors, detected while performing operations, to be printed. From the HCD LDAP backend point of view, LDAP_DEBUG_ERROR can be switched on regularly without significant performance loss.

LDAP_DEBUG_TRACE causes all important program operations to be printed. From the HCD LDAP backend point of view, LDAP_DEBUG_TRACE should only be used when reproducing error situations for problem determination.

If you have general problems getting the IBM Tivoli Directory Server for z/OS, RACF Backend, or any other plug-ins running, leave out all HCD LDAP backend definitions from the started task and configuration files (ds.conf). If the server and plug-ins still will not run, contact the IBM service.

If everything functions correctly without the HCD LDAP backend and abends with the HCD LDAP backend, then request a dump and check there for further information.

If the dump indicates that functions in DLL GLDSP31 or GLDCLDAP are missing, then maybe the IBM Tivoli Directory Server for z/OS functionality is not available.

HCD LDAP backend abnormal termination

There are two different locations at which the HCD LDAP backend might terminate abnormally:

- The IBM Tivoli Directory Server for z/OS address space
- The HCD instance address space

The way to find out more information about the abnormal termination depends on in which of these locations it terminated see "IBM Tivoli Directory Server for z/OS address space" or "HCD instance address space."

In general, whenever the HCD LDAP backend or parts of it terminate abnormally, some global resources may remain in memory. These resources are so called POSIX message queues and are used for the Inter Process Communication (IPC) between the HCD LDAP backend and the HCD instances. They have to be removed from the system manually.

To remove POSIX message queues from the system proceed as follows:

1. Look into the IBM Tivoli Directory Server for z/OS started task log to find the identifiers (ID's) of all message queues allocated: The HCD LDAP backend prints out all ID's using the message CBD0004I. Note that you have to find all of these CBD0004I messages.
2. Use the UNIX System Services command line tool `ipcs` to list all the POSIX message queues in memory.
3. Find those identifiers in the list which correspond to POSIX message queues belonging to the abnormally terminated HCD LDAP backend.
4. Remove the appropriate message queues using UNIX System Service command line tool `ipcrm`.

IBM Tivoli Directory Server for z/OS address space: In general, diagnostic output can be found in the IBM Tivoli Directory Server for z/OS started task log. Investigate this log to find information, about the system or user abend code, the reason code, and additional messages and follow the instructions in "HCD abnormal termination" on page 470.

For more information on problem determination, refer to *z/OS V2R2 Problem Management*. Read the information on how to obtain a dump of the abnormally terminated program and determine whether the problem was caused by the HCD LDAP backend or by any other program in the IBM Tivoli Directory Server for z/OS address space.

HCD instance address space: Since an HCD instance resides in a different address space (created with a call to `spawn()`) to that of the corresponding IBM Tivoli Directory Server for z/OS, it is not permitted to print its output to the IBM Tivoli Directory Server for z/OS started task log. Instead, an HCD instance creates an HFS file containing diagnostic output. The name and location of this file follows the conventions described in the publication *z/OS Language Environment Debugging Guide* in the unit "Using Language Environment Debugging Facilities". Note that the environment variable `_CEE_DMPTARG` is set to `/tmp` by default.

Example: Assume that a language environment dump of the HCD instance with the process ID 197 is taken at 05:55:01 pm on September 18, 1998. The file containing the dump will be called `CEEDUMP.19980918.175501.197` and it will be located in the HFS directory `/tmp`.

Adverse LDAP return code from the HCD LDAP Backend

Each LDAP client's request is answered by the IBM Tivoli Directory Server for z/OS with an LDAP response containing a return code for the request. Such a return code can be either good (LDAP_SUCCESS) or bad (e.g. LDAP_UNWILLING_TO_PERFORM). These numeric return codes can be transformed into text messages (e.g. "DSA is unwilling to perform.") using the LDAP client API.

Return codes such as LDAP_OPERATIONS_ERROR, LDAP_NO_MEMORY, or LDAP_LOCAL_ERROR typically indicate a problem within the IBM Tivoli Directory Server for z/OS, the HCD LDAP backend, or the LDAP client. If you assume the problem to be located in the IBM Tivoli Directory Server for z/OS or the HCD LDAP backend, look into the IBM Tivoli Directory Server for z/OS started task log and search for error messages.

Return codes such as LDAP_INAPPROPRIATE_MATCHING, LDAP_INVALID_DN_SYNTAX, or LDAP_UNWILLING_TO_PERFORM typically indicate an incorrect LDAP request. Check if your request conforms with the appropriate constraints. You may also check the IBM Tivoli Directory Server for z/OS started task log for error messages.

Error in the HCD LDAP backend

If the HCD LDAP backend message CBD0009E is printed to the IBM Tivoli Directory Server for z/OS started task log, an error situation was encountered inside the HCD LDAP backend while performing an operation. Switch on IBM Tivoli Directory Server for z/OS debug level LDAP_DEBUG_ERROR and reproduce the problem. The log will contain more information concerning the reason of the error. You may additionally specify debug level LDAP_DEBUG_TRACE to get detailed informations about the operations performed by the HCD LDAP backend.

The following illustrates two typical error situations with advice on how to check for these problems.

HCD Instance Startup Fails: If an HCD instance cannot be started correctly, this will not lead to error messages in the IBM Tivoli Directory Server for z/OS started task log or to bad return codes for the LDAP client in all cases.

Consider the following possible scenario: The IBM Tivoli Directory Server for z/OS is started together with the HCD LDAP backend which, in turn, launches at least one HCD instance. All launched HCD instances fail to start. As the HCD LDAP backend has no indication that the HCD instances failed to start, it will wait forever for an acknowledgement from the HCD instances. Incoming requests will be scheduled for an HCD instance and will be blocked up until an acknowledge arrives. In this way, the requests will never be performed and no LDAP response will be sent to the requesting LDAP client.

From the LDAP client's point of view: If the very first request for the HCD LDAP backend is pending for a long time, the HCD instances might have failed to start up. In general, if one HCD instance fails to start up, all HCD instances will fail.

To check whether an HCD instance failed to start, look in the IBM Tivoli Directory Server for z/OS started task log: HCD LDAP backend message CBD0007I will indicate how many HCD instances will be started. If all of these HCD instances are able to start up correctly, you will find an appropriate number of CBD1002I messages. If you cannot find them it's most probable that the HCD instances failed to start up.

HCD instance user ID switch fails: If you have set up the IBM Tivoli Directory Server for z/OS or the HCD LDAP backend incorrectly, the HCD instances might have problems switching their User ID appropriately. It might be possible that the User ID switch fails completely, or only for one or more specific User IDs.

If you assume that LDAP requests can't be performed because of a failed User ID switch, you can check your assumption as follows: Search for the CBD0009E message in the IBM Tivoli Directory Server for z/OS started task log. If you can find it switch on IBM Tivoli Directory Server for z/OS debug level LDAP_DEBUG_ERROR and search for message EDC5139I. This message indicates that the User ID switch has failed.

Finally, check the setup of your HCD LDAP backend.

Diagnostic information and tools

The information and tools described in this section help you to diagnose system problems.

HCD messages

In case of an error, HCD issues messages. Depending on what you are currently doing, the messages are written:

- To the terminal as a single message
- To the terminal in a message list
- In a message log
- In a migration log
- In the output of a batch job

Terminal messages

User-errors, such as erroneous syntax entry and contextually wrong definitions, are handled by the dialog at the time of data entry. That is, the dialog displays messages at the terminal and the user can take corrective action immediately.

Some operations produce multiple messages. In this case, HCD displays a message list. You can save the displayed messages from the message list into the message log. See "Message lists" on page 62 on how to work with message lists.

Message log

Errors that are of low interest for the end user, such as incomplete UIMs during initialization, are only written to the message log. The user will be informed about this when leaving the dialog or switching to another IODF. Only in critical situations (for example, when the message log is not available), will the messages be written into the ISPF list data set. If this, however, also fails, the message will be written into the operating system log.

To see a message in the message log, issue the SHOWMSG command or use the *View message log* pull-down choice from the *Query* action bar on any action list panel.

Migration log

HCD maintains a migration log that contains messages issued by the migration process. You can view this migration log through ISPF.

Table 51 shows where you can find messages while working with HCD.

Table 51. Where to find HCD messages

Mode of Operation	You Find the Message
Dialog Mode	On the terminal In the message log
Batch mode	In the message log, that is the data set allocated with ddname HCDMLOG In the output (SYSPRINT) of the batch job ¹
Migration of input data sets	In the migration log, that is the data set allocated with ddname HCDPRINT
IPL	Trapped by IPL. A wait state code is issued.
HCD LDAP backend	In the LDAP response, which the IBM Tivoli Directory Server for z/OS sends to the LDAP client. In the IBM Tivoli Directory Server for z/OS started task's log.

Note: ¹ The output shows the message number and text; for explanations refer to *z/OS and z/VM HCD Messages*.

LDAP response messages

If HCD is used as part of the HCD LDAP backend, HCD will issue messages if problems occur. These messages can be made visible in the following way: In general, LDAP clients send their requests to the IBM Tivoli Directory Server for z/OS and get an LDAP response for each request. If an LDAP client requests an operation from HCD via the HCD LDAP backend and HCD encounters any problems while working on the request, HCD issues one or more text messages. These messages are sent back to the LDAP client as a part of the LDAP response and can be extracted using the LDAP client API.

The command line utilities (e.g. `ldapsearch` or `ldapmodify`) which are part of the IBM Tivoli Directory Server for z/OS show the messages contained in the LDAP response as "additional info".

IBM Tivoli Directory Server for z/OS started task log

The HCD LDAP backend runs as a part of the IBM Tivoli Directory Server for z/OS and all HCD LDAP backend messages are printed to the same location as the IBM Tivoli Directory Server for z/OS messages. The IBM Tivoli Directory Server for z/OS runs as started task for the HCD LDAP backend and the messages from the server and all of its backends are printed to the IBM Tivoli Directory Server for z/OS started task log.

The HCD LDAP backend and its HCD instances print some informational and error messages to the IBM Tivoli Directory Server for z/OS started task log. If the LDAP server debug is switched on (i.e. debug level is not `LDAP_DEBUG_OFF`), all diagnostic output is also printed to the started task log. The regular HCD LDAP backend messages and the additional debug output can be used to verify that the HCD LDAP backend is working correctly.

HCD trace facility

The output of the HCD trace facility provides information to locate internal HCD problems. It helps the IBM program system representative to identify the cause of a failure.

Data Set

The trace records generated by HCD are recorded in the trace data set.

The trace data set must be preallocated with a ddname of HCDTRACE. If the trace data set is not allocated when HCD is invoked, no tracing takes place. The default CLIST that is provided with HCD allocates a trace data set with the name HCD.TRACE, prefixed by your user ID.

Trace records

The trace records show the control flow within the various HCD modules. Trace information is written into the trace data set:

- Whenever a module (CSECT) gets control.
In this case, the passed parameter list is recorded together with the name and description of the invoked module.
- Whenever a module (CSECT) returns to its calling routine.
The passed parameter list containing the return and reason codes is recorded.

When HCD is invoked, HCD overwrites the existing trace with the new trace data. If you want to retain the existing data, you have to make sure that the data is saved.

Figure 244 on page 490 is an example of a trace output. The following explains the records you see:

```
17:00:13 96-08-01 Trace started
```

The time stamp shows when the trace facility was started. This record is useful to identify the trace data sets when multiple traces are produced on the same day.

```
PUSH CBDMSMSG - Message Routine 96214 HCS6031 17001352 08472028
```

This line indicates that control has been passed to another module at a certain time. In this example, the module named CBDMSMSG with service level '96214 HCS6031' received control at 17:00:13.52 at storage address X'08472028'. The records also give a short description of what the module does, and the parameter list that is passed to the called module.

```
CBDMSMSG Message destination: Screen
```

This trace entry is written by the called module.

```
POP CBDMSMSG - Message Routine 17001353
```

Indicates that control from the module named CBDMSMSG is returned to the calling routine at 17:00:13.53. Also, the passed parameter list is shown again, but now the parameter list contains the return/reason code indicating how successful the requested function was.

```
17:01:24 96-08-01 Trace stopped
```

The last entry in the trace output is a time stamp that indicates end of tracing.

```

17:00:13 96-08-01 Trace started.
.
.
PUSH CBDMSMSG - Message Routine 96214 HCS6031 17001352 08472028
00000000 D4E2C740 C3C2C4C1 F1F2F6C9 C9000100 *MSG CBDA126II""*
00000010 00000000 00000000 00000000 00000000 *""""""""""*
00000020 00000000 00000000 00000000 00000000 *""""""""""*
00000030 00000000 00000000 00000000 00000000 *""""""""""*
00000040 00000000 00000000 00000000 00000000 *""""""""""*
00000050 00000000 00000000 00000000 00000000 *""""""""""*
00000060 00000000 00000000 00000000 00000000 *""""""""""*
00000070 00000000 00000000 00000000 00000000 *""""""""""*
CBDMSMSG Message destination: Screen
00000000 E3D9C1C3 C5408396 94948195 8440A681 *TRACE command wa*
00000010 A2408183 838597A3 85844B *s accepted. *
POP CBDMSMSG - Message Routine 17001353
00000000 D4E2C740 C3C2C4C1 F1F2F6C9 C9000100 *MSG CBDA126II""*
00000010 00000000 00000000 00000000 00000000 *""""""""""*
00000020 00000000 00000000 00000000 00000000 *""""""""""*
00000030 00000000 00000000 00000000 00000000 *""""""""""*
00000040 00000000 00000000 00000000 00000000 *""""""""""*
00000050 00000000 00000000 00000000 00000000 *""""""""""*
00000060 00000000 00000000 00000000 00000000 *""""""""""*
00000070 00000000 00000000 00000000 00000000 *""""""""""*
.
.
17:01:24 96-08-01 Trace stopped.

```

Figure 244. Example: Trace output

Figure 245 on page 491 is an extract of a trace output when an abend occurred. The following explains the record in the example.

ESTAE: The entries show information that was recorded by the HCD ESTAE routine. Entries under *Diagnostic stack* list the modules that were executing when the abnormal termination occurred, together with the service levels. The first entry names the module that ended abnormally; in this example, the module is CBDMYCCB. The entries also show the control flow between the various HCD modules and the load address of the module. In this example:

- Module CBDMYDCC called module CBDMYCCB loaded at 212E0EF8.
- Module CBDMGDIA called module CBDMYDCC loaded at A12D4000.
- Module CBDMGHCP called module CBDMGDIA loaded at A1846B50.

```

*****
*           E S T A E                               *
*           System abend code   : 0C4                *
*           Reason code        : 00000011           *
*           HCD version        : z/OS 2.3 HCD        *
*           Diagnostic stack    : CBDMYCCB 15238    HCS77B0 212E0EF8 *
*                               CBDMYDCC 15237    HCS77B0 A12D4000 *
*                               CBDMGDIA 15194    HCS77B0 A1846B50 *
*                               CBDMGHCP 15197    HCS77B0 800695E0 *
*                               *                    *
*           Module name        : CBDMGHCP           *
*           Entry point address: 000695E0           *
*           PSW                : 078D1000 A12FB2AE  *
*                               *                    *
*           R0 00000030 R1 21AD3000 R2 21AD3194 R3 00002F10 *
*           R4 21305608 R5 21305608 R6 21306607 R7 212FB1CF *
*           R8 21307606 R9 21308605 R10 00000450 R11 21AD35E4 *
*           R12 A12FA1D0 R13 21305AD4 R14 21ADF060 R15 21B5BF10 *
*****

```

Figure 245. Example: Trace output in case of Abend

Activating the trace

The trace can be activated either:

- Dynamically by using the TRACE command in the HCD dialog. The command is optional, and can be entered whenever a displayed panel has a command line. For information on the TRACE command, refer to “TRACE command.”
- By invoking HCD (for details see Chapter 13, “How to invoke HCD batch utility functions,” on page 327) with the TRACE option specified in the passed parameter string (this is done automatically by the entry in the CLIST).

```

//BWINJOB JOB (3259,RZ-28), 'BWIN', NOTIFY=BWIN, CLASS=A,
//          MSGCLASS=Q, MSGLEVEL=(1,1), REGION=4M
//REPORT1 EXEC PGM=CBDMGHCP,
//          PARM= 'TRACE, REPORT, CSMEN, PROC1, PART1, MVS1, 00'
          .
          .
//HCDTRACE DD DSN=BWIN.HCD.TRACE, DISP=OLD
//

```

- By specifying the TRACE command in the HCD profile, for example,


```
TRACE ON RESET,HOM,V,R LEVEL=255
```

The tracing stays active until either turned off by the TRACE command, or until HCD terminates.

TRACE command

The TRACE command activates and deactivates the HCD trace facility. The command allows you also to limit the detail of data written into the trace data set by requesting that only certain functions and details should be traced.

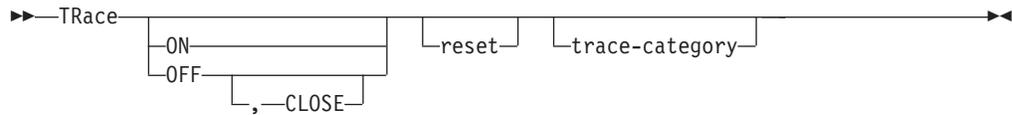
The TRACE command can be entered on any HCD panel showing a command line. The command can also be specified in the HCD profile data set. It is, however, not shown in the *HCD Profile Options* dialog.

For HCD to write the output to the trace data set, ON must be specified with at least one trace category (or you must have invoked HCD with the TRACE

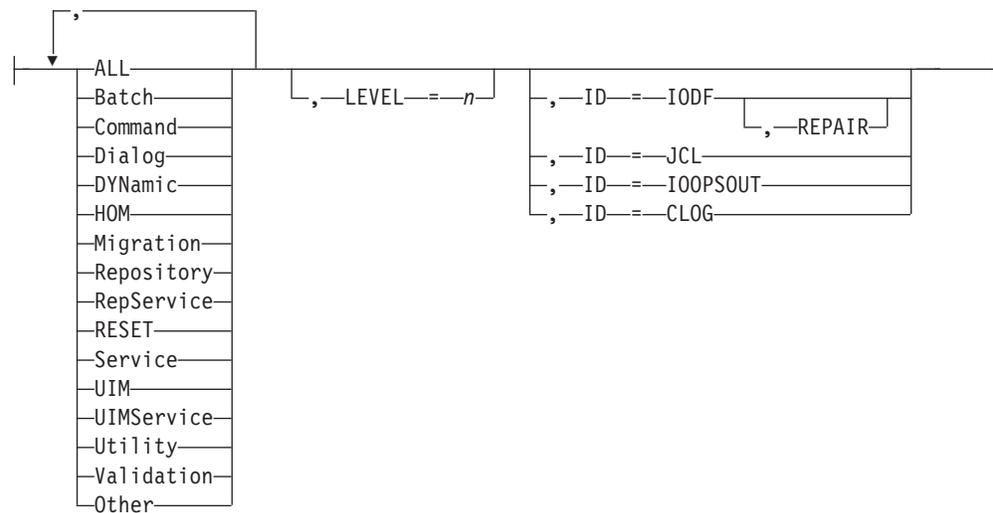
parameter, see “Activating the trace” on page 491). To view the trace output, you have to close the trace data set first. You can do this by either leaving HCD or by entering the command TRACE OFF,CLOSE.

The format of the command is as follows:

TRACE Command



trace-category:



Notes:

1. You may abbreviate some of the keywords. The characters you have to use are indicated by uppercase (you must then omit lowercase). For example RepService may be abbreviated as RS.
2. At least one parameter must be specified with TRace.

ON Starts the trace facility.

OFF Stops the trace facility.

CLOSE
Closes the trace data set.

Note: ON, OFF and CLOSE can not be used if the profile is allocated by the HCD dispatcher and used by HCM.

reset Reset all currently active categories, LEVEL and ID. If not using RESET, HCD will remember the trace categories specified with previous trace commands and adds them up. If combined with trace categories, it should be specified before the categories, because the keywords and categories are evaluated in sequence. RESET at the end would clear the specified trace category.

trace category

Specifies the functional scope to be traced:

ALL Traces for all categories except DEBug and RepService. To have a complete trace, use categories ALL, DEBug, RepService.

Batch Trace all batch routine.

Command

Trace all command routines.

DEBug

Traces internal services (not included, when ALL is specified).

Dialog

Trace all dialog routines.

DYNamic

Trace all dynamic routines.

HOM Trace all object management routines.

Migration

Trace all migration routines.

Repository

Trace all repository main routines.

RepService

Trace all repository service routines. (Not included, when ALL is specified).

Service

Trace all service routines.

UIM Trace all UIM routines.

UIMService

Trace all UIM service routines.

Utility Trace all utility routines.

Validation

Trace all validation routines.

Other Trace all other not yet mentioned routines.

LEVEL=*n*

Assigns a level of detail to the functions to be traced, where *n* is a decimal number ranging from 0 to 255. If the option is omitted, the default level of 5 is assumed. The TRACE option described in "Input parameter string" on page 328 is equivalent to the command TRACE ON,ALL,LEVEL=255.

ID=IODF

Writes an IODF dump into the trace data set. This parameter cannot be specified in the HCD profile. If you have a consistent IODF, an output in the trace data set is only shown when you set LEVEL=128 or higher. Otherwise, an output is only shown if the IODF contains defects.

REPAIR

Removes detected errors in the work IODF and reports corrections in the trace data set. Before you use the REPAIR option, you must set the work IODF in update mode.

ID=JCL

Writes into the trace data set all statements generated when action *Transmit configuration package* is invoked from the HCD dialog.

ID=IOOPSOUT

Writes all responses of I/O Operations IHVAPI2 calls into the trace data set. These are the results of I/O Operations query requests.

ID=CLOG

Writes the contents of the change log file into the HCD trace data set. You should use this option together with LEVEL=8. This parameter cannot be specified in the HCD profile.

Trace command via HCD profile

- If HCD is started with the TRACE keyword, (for example, either started via HCM with the *HCD Trace* box selected in the HCM login dialog, or via the IBM Tivoli Directory Server for z/OS configuration file), initially *all* the trace categories will be traced. After the HCD profile has been read, however, the TRACE parameters there may modify the TRACE behavior.
- The TRACE parameters set in the HCD profile will also influence the TRACE behavior if you activate the tracing in HCM at a later time.
- The TRACE parameters set in the HCD profile will determine the contents of the trace for the rest of the session.
 - If you use the RESET option followed by trace categories *cat1,cat2,...,catn*, then only the categories *cat1,cat2,...,catn* will be considered. The RESET option must be the first option because the trace categories are additive (LEVEL is set to 0).
 - Specifying the keyword off will terminate the startup trace.
 - Specifying the keyword on will start the HCD trace (if not already started) and will invoke the trace parameters of the TRACE statement.
 - Specifying LEVEL=*n* will set the level of trace detail. If the LEVEL parameter is not set, then the trace will use the default level of 5.
- If no categories are set explicitly, then all trace categories will be active.

IODF dump

Use the ID=IODF parameter of the TRACE command to produce an IODF dump. This command goes through your IODF, checks it for corrupted data, and writes all records and defects into the trace data set. If you have a consistent IODF, you must set the LEVEL parameter to LEVEL=128 or higher to get an output. Otherwise, an output is only shown if the IODF contains defects.

If your IODF has defects, error message CBDA999I 'Defect(s) detected in IODF xxx' is displayed, and message CBDA099I is written into the message log data set. In addition, the trace data set records defects with the string 'Error:' followed by the reason. You can locate the reported defects by searching to that string in the trace data set.

If no defects are detected in the IODF, message CBDA126I 'TRACE command was accepted' is given.

If you cannot invoke HCD, and therefore, cannot use the TRACE command any longer, use a JCL stream for producing a dump. Figure 246 on page 495 is an example of the JCL stream for producing an IODF dump.

Make changes to the entries according to your installation requirements.

```
//WAS$IODF JOB '3259,BOX01,S=C','SMITH',MSGLEVEL=(1,1),
//          NOTIFY=WAS,CLASS=A,MSGCLASS=Q,REGION=4096K
//PRTVSAM EXEC PGM=IDCAMS
//INPUT DD DSNAME=WAS.IODF02.WORK,DISP=SHR
//OUTPUT DD DSNAME=WAS.IODF02.DUMP,UNIT=SYSALLDA,
//          SPACE=(CYL,(20,10)),
//          DCB=(LRECL=125,BLKSIZE=629,RECFM=VBA),
//          DISP=(NEW,CATLG)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
PRINT -
        INFILE(INPUT) -
        DUMP -
        OUTFILE(OUTPUT)
/*
//
```

Figure 246. Example: JCL stream for producing an IODF dump

Repair an IODF

If your IODF contains defects, some of them can be repaired with the REPAIR option of the TRACE ID=IODF command.

First, your IODF must be set in update mode to correct defects. You can do this, for example, by changing a processor description field in your work IODF. A production IODF cannot be repaired.

To repair defects in your work IODF, add the REPAIR option to the TRACE ID=IODF command: TRACE ON,REPAIR,ID=IODF.

Each corrected defect is recorded in the trace data set with the string Defect has been corrected.

If a defect has been repaired, message CBDA998I, Defect(s) detected in IODF xxx. Repair action performed is issued. Repeat the TRACE ID=IODF command to check whether all defects could be corrected.

An IODF that has been enabled for multi-user access cannot be repaired. You first must disable it for multi-user access before it can be set into update mode and be repaired.

If defects in the IODF cannot be repaired via the REPAIR option, you may have to rebuild the IODF. You accomplish this task by exporting the configuration data from the IODF via dialog option 2.10 *Build I/O configuration data*. You can then migrate this data to a new IODF:

1. Export configuration types Processor, Operating System and Switch with configuration ID * into three different output data sets (see “Build I/O configuration data” on page 226). This will generate I/O configuration statements for all processor, operating system and switch configurations. You can set the profile option SHOW_CONFIG_ALL to YES to also generate configuration statements for unconnected control units and devices in addition to those for switches (see “Export/import additional configuration objects” on page 28).

2. Successively migrate the processor, OS configuration and switch data into a new IODF (see “Migrate I/O configuration statements” on page 331). You must use batch jobs for this purpose, since in dialog mode, HCD does not support the configuration ID *.
3. Verify - for example using compare reports - that the new IODF includes all data. If necessary, add any missing items manually.
4. Check that the IODF is free from defects.

MVS dumps and traces

To aid in diagnosing problems, z/OS automatically provides messages and error records, and on request dumps and traces. HCD uses those services to record errors. For information about:

- Dumps and traces, refer to *z/OS MVS Diagnosis: Tools and Service Aids*.
- Using the diagnostic information, refer to *z/OS V2R2 Problem Management*.

IPCS reports

z/OS allows you to format dumps into diagnostic reports. To produce the reports, use the Interactive Problem Control System (IPCS).

For information, refer to:

- *z/OS MVS IPCS User's Guide*
- *z/OS MVS IPCS Commands*
- *z/OS MVS IPCS Customization*

Searching problem reporting data bases and reporting problems

Search arguments are used to search problem reporting data bases. If the problem being diagnosed was already reported and the symptoms entered into the data base, the search will produce a match.

To perform a search, do the following:

1. Analyze the problem reporting data base and develop a search argument using the information provided in the boxes labeled *Search Argument*.
2. Complete the digits (such as ccc, nnn, hhh) according to the applicable conditions. For example, if the message CBDA099I was received, the developed search argument for *message identifier* would be: MS/CBDA099I. An example is shown in Table 31 on page 471.
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 in the *Problem Data* tables. An example is shown in Table 32 on page 471.

For more detailed information on these steps, refer to *z/OS V2R2 Problem Management*.

Sending an IODF to a different location

There may be situations, in which an IODF is to be transferred to a different location or system. Usually, the HCD Export/Import Utility can be used to transmit the IODF to the desired target destination. Sometimes, however, there are situations, where this may not be possible (e.g. a direct connection does not exist).

The following steps describe a simple method in which you can transfer your IODF data from one z/OS host to any other z/OS host even if a direct transmission path not available.

1. Examine the size of the IODF, which is to be transferred. Check for the number of allocated 4K blocks in the HCD dialog: *Maintain I/O definition files* ---> *View I/O definition file information*.
2. Use the HCD Export function, found under the HCD dialog *Define, modify, or view configuration data, Export I/O definition file* in the HCD dialog. Send the IODF to your own user ID; that is, to the user ID of the host on which you are currently working. HCD will export the IODF data as a sequential data set to your own user ID. If you specify an asterisk (*) for the target user ID and target node ID, the sequential data set is not transmitted but rather written directly to the data set *user.EXPORTED.IODFnn.xxxx*. In this case, you can skip step 4.
3. Exit the HCD dialog.
4. Use the TSO RECEIVE command to retrieve the IODF data from your internal reader. Per default, you will get a sequential data set *user.EXPORTED.IODFnn.xxxx*. This data set has the record organization FB and LRECL=BLKSIZE=4096 and the number of blocks as mentioned under step 1).
5. Download this data set to your workstation. It is important that you ensure that the download is in *binary* mode.
6. To save storage resources, you may consider compressing (zipping) the downloaded file on the workstation.
7. Now you can transfer the IODF to a different workstation/location. Once the IODF data has arrived at the target workstation, you have to uncompress (unzip) the file if it has been compressed for transportation.
8. On the target z/OS host pre-allocate a data set into which the IODF data is to be uploaded. It must be a sequential data set with FB, BLKSIZE=LRECL=4096 and the number of blocks must be the number of allocated blocks of the original IODF (see step 1).
9. Upload the IODF data from the workstation to the z/OS host in binary mode into the pre-allocated sequential data set.
10. Use the HCD Import function, found under the HCD dialog *Define, modify, or view configuration data, Import I/O definition file* and specify the sequential data set to be imported into an IODF data set using the name of your choice.

Appendix D. HCD object management services

Programming Interface information

The HCD object management services (HOM) provide an application programming interface for retrieving data from the IODF, such as switch data, device type, or control unit type. The programs requesting the services cannot run in APF-authorized state.

The mapping macros CBDZHRB, CBDZHIEX, CBDZHOEX, and CBDZHCEX (see “Data input and output areas” on page 500 and “Request block (HRB)” on page 501) are not available as source code. The macros are listed in *MVS Data Areas Volume 1* available from the z/OS Internet Library, and must be coded by the application writer.

How to invoke the HOM services

Programs can invoke the services from the HCD routine CBDMGHOM. An application issuing a request must have its own copy of the CBDMGHOM routine dynamically loaded or linked. For every request, the application must use the same HRB address for each call to CBDMGHOM and must pass the parameters shown in Table 52 using standard linkage conventions.

Table 52. Used registers and passed parameters

Register	Contents
0	Undefined

Table 52. Used registers and passed parameters (continued)

Register	Contents
1	<p>Address of five-word parameter list:</p> <ol style="list-style-type: none"> Address of request control block (HRB) 4-byte field containing the address of the request block. The request block contains the function, the object to which the function is applied, and qualifiers, attributes, and parameters. See "Request block (HRB)" on page 501 for more details. Address of (pointer to input data or zero) 4-byte field containing the address of the address of the data input block if the request requires input. It is required on a HRB_SETUP request. See "Data input and output areas" for more details. Address of (length of input data or zero) 4-byte field containing the address of the fullword fixed binary integer containing the length of the input data. It must correspond to the exact length of the data contained in the data-input block, that is, no trailing or intermediate blanks are allowed. Address of (pointer to output data or zero) 4-byte field containing the address of the address of the data output block if the request returned output. It is required on HRB_DGET and HRB_MGET requests to obtain the data and messages from the API. See "Data input and output areas" for more details. Address of (length of output data or zero) 4-byte field containing the address of the fullword fixed binary integer containing the length of the output data. <p>The parameters must be coded in the order shown. Only the first parameter (address of request block) is mandatory. The others are optional and depend on the type of request, as shown in Table 54 on page 502. If you omit an optional parameter, you must specify a zero instead.</p>
2-12	Undefined
13	Address of 18 word save area
14	Return address
15	Entry point address

Note: The service supports calls for both 24-bit and 31-bit addressing mode.

Data input and output areas

Data input and output areas must be contiguous areas of main storage allocated by the application in private storage and freed later on.

The areas have no header section, that is, the data starts at the first byte of the area and continues without gaps. The data contained in these areas are the interface records, which are described in the mapping macros CBDZHOEX and CBDZHIEX in MVS Data Areas Volume 1.

For the GET request, the output area might contain the definition of multiple objects on return, whereas the input area normally contains only one object.

Issue a HRB_DGET function to get the provided output data. The size of the output data is returned by the previous GET request. The application is responsible to allocate the correct output size. If the data does not fit into the size allocated by the application for the output area, the data will be truncated.

Request block (HRB)

The HOM request block (HRB) you have to set up is described fully as mapping macro CBDZHRB in *MVS Data Areas Volume 1* available from the <http://www.ibm.com/systems/z/os/zos/bkserv/>. Table 53 summarizes the request block names and constants you can specify for the functions shown in Table 54 on page 502.

On input, this block contains the detailed request to the HOM services. On output, it contains the data requested, messages, return codes, and reason codes.

The field HRB_OBJECT with all its subfields describes the object that should be processed on the request to the API.

The object code HRB_OBJ_CODE must be coded for every request, because it identifies the class of objects that are subject of the actual request.

The constants and flags required to describe the objects are contained in CBDZHCEX, which is documented in *MVS Data Areas Volume 1*.

Table 53. Summary of Request Block Names and Related Constants

Name	Constants	Description
HRB_SDESC	HRB_SDESC_C	Request block storage descriptor. Required for all requests.
HRB_LENGTH		Length of the request block HRB.
HRB_USE_IODF		Name of the IODF to be used for the request.
HRB_FUNCTION	HRB_SETUP HRB_OPEN HRB_GET HRB_ACT_STATUS HRB_DGET HRB_MGET HRB_CLOSE HRB_TERMINATE	Function code
HRB_OBJ_CODE	HRB_HCD	Required for SETUP and TERMINATE.
	HRB_IODF	Required for OPEN and CLOSE.
	HRB_PROCESSOR	Required for processor.
	HRB_CSS	Required for channel subsystem.
	HRB_PCU	Required for physical control unit.
	HRB_DEVICE	Required for device.
	HRB_SWITCH	Required for switch.
	HRB_CHANNEL	Required for channel path.
	HRB_DATA	Required for DGET.
	HRB_MESSAGE	Required for MGET.
	HRB_IODF	Required for ACT_STATUS.
HRB_OBJ_NAME		May be used to specify the name and number of an object. For devices, the number includes the suffix.
HRB_OBJ_NR		

Table 53. Summary of Request Block Names and Related Constants (continued)

Name	Constants	Description
HRB_Q_CODE	HRB_PCU HRB_DEVICE HRB_SWITCH	May be used together with HRB_PROCESSOR to specify that the processor data is qualified by the control unit, device, or switch.
	HRB_PROCESSOR	May be used together with HRB_CHANNEL to determine the kind of channel path data.
HRB_Q_NAME		May be used to specify the qualifier name and number for composite names.
HRB_Q_NR		
HRB_REQ_MODE	HRB_MODE_ID	Gets objects starting with the ID specified. The ID of the object must be set in HRB_OBJ_NR or HRB_OBJ_NAME. The HRB_RANGE_VALUE must not be zero.
	HRB_MODE_FIRST	Gets first object in the defined scope.
	HRB_MODE_LAST	Gets last object in the defined scope.
	HRB_MODE_ALL	Gets all objects in the defined scope.
	HRB_MODE_CHAIN	Gets all objects within the chain defined by the given object, for example, all devices of a multi-exposure device.
HRB_RANGE_VALUE		May be used to specify the number and direction of objects to be processed (positive number = subsequent objects; negative number = preceding objects).
HRB_TRACE	HRB_YES	The request is traced. Make sure that the trace data set is allocated with a DD name of HCDTRACE.
HRB_RESULT		The subfields of HRB_RESULT contain the output of the request, such as the data requested, the size of the output data, or return codes.

Functions

The HCD application programming interface provides the functions described in Table 54. The functions are listed as you need them while requesting data from the HCD HOM services. The constants you have to specify for HRB_FUNCTION and HRB_OBJ_CODE are included in the figure.

Table 54. Functions provided by the HOM services

Task	Fields in Request Block (HRB)		Explanation
	HRB_FUNCTION=	HRB_OBJ_CODE=	
1. Set up the connection to the HCD API			
Setup connection	HRB_SETUP	HRB_HCD	Establish the HCD environment by passing the setup function in the request block. Input: HCD session interface (HSI) record.
2. Open the IODF			
Open IODF	HRB_OPEN	HRB_IODF	Open an IODF, for which you have read authority, by passing the HRB_OPEN function in the request block.
3. Request data for HCD objects			

Table 54. Functions provided by the HOM services (continued)

Task	Fields in Request Block (HRB)		Explanation
	HRB_FUNCTION=	HRB_OBJ_CODE=	
Get Processor	HRB_GET	HRB_PROCESSOR	Issue a request with the GET function to retrieve data from the IODF. The request returns the address and the length of the data output block. Issue a request with the HRB_DGET function to obtain the retrieved information. Note: Issue HRB_GET and HRB_MGET before requesting additional data, because the new request deletes all data and messages from the previous request.
Get Channel Subsystem	HRB_GET	HRB_CSS	
Get Channel Path	HRB_GET	HRB_CHANNEL	
Get Switch	HRB_GET	HRB_SWITCH	
Get Physical Control Unit	HRB_GET	HRB_PCU	
Get Device	HRB_GET	HRB_DEVICE	
Get Activation Status	HRB_ACT_STATUS	HRB_IODF	
4. Get the data from the previous GET request			
Data Get	HRB_DGET	HRB_DATA	Issue the request with the address and length of the output area to obtain the data retrieved with the previous GET function. Output: Interface record for the object.
Message Get	HRB_MGET	HRB_MESSAGE	Check the return and reason code of the previous GET request. If the return code is warning, error, or severe, issue the request with the address and length of the output area to obtain the messages. Output: Message interface (MSI) record.
5. Close the IODF			
Close IODF	HRB_CLOSE	HRB_IODF	When you do not need the IODF anymore, close the IODF by issuing a request with the close function.
6. Terminate the connection to the HCD API			
Terminate connection	HRB_TERMINATE	HRB_HCD	When you do not need the HOM services anymore, terminate the connection by issuing a request with the terminate function.

Example

The example shows how to get a range of 20 devices, starting with ID X'414' and connected to control unit X'21'. The example is shown in pseudo-code because the actual syntax and declarations depend on the programming language used.

```

...
HRB_SDESC      = HRB_SDESC_C
HRB_LENGTH     = length-of-HRB
HRB_FUNCTION   = HRB_SETUP
HRB_OBJ_CODE   = HRB_HCD
Load or link CBDMGHOM with: (HRB,HSI-address,HSI-length,0,0)

...
HRB_SDESC      = HRB_SDESC_C
HRB_LENGTH     = length-of-HRB
HRB_USE_IODF   = IODF-name
HRB_FUNCTION   = HRB_OPEN
HRB_OBJ_CODE   = HRB_IODF
Load or link CBDMGHOM with: (HRB,0,0,0,0)

```

```

...
HRB_SDESC      = HRB_SDESC_C
HRB_LENGTH     = length-of-HRB
HRB_USE_IODF   = IODF-name
HRB_FUNCTION   = HRB_GET
HRB_OBJ_CODE   = HRB_DEVICE
HRB_OBJ_NR     = X'0414000'
HRB_Q_CODE(1)  = HRB_PCU
HRB_Q_NR(1)    = X'00000021'
HRB_REQ_MODE   = HRB_MODE_ID
HRB_RANGE_VALUE = X'00000014'
Load or link CBDMGHOM with: (HRB,0,0,0,0)
...
HRB_SDESC      = HRB_SDESC_C
HRB_LENGTH     = length-of-HRB
HRB_USE_IODF   = IODF-name
HRB_FUNCTION   = HRB_DGET
HRB_OBJ_CODE   = HRB_DATA
Load or link CBDMGHOM with: (HRB,0,0,DVI-address,DVI-length)
....
HRB_SDESC      = HRB_SDESC_C
HRB_LENGTH     = length-of-HRB
HRB_USE_IODF   = IODF-name
HRB_FUNCTION   = HRB_MGET
HRB_OBJ_CODE   = HRB_MESSAGE
Load or link CBDMGHOM with: (HRB,0,0,MSI-address,MSI-length)
...
HRB_SDESC      = HRB_SDESC_C
HRB_LENGTH     = length-of-HRB
HRB_USE_IODF   = IODF-name
HRB_FUNCTION   = HRB_CLOSE
HRB_OBJ_CODE   = HRB_IODF
Load or link CBDMGHOM with: (HRB,0,0,0,0)
...
HRB_SDESC      = HRB_SDESC_C
HRB_LENGTH     = length-of-HRB
HRB_FUNCTION   = HRB_TERMINATE
HRB_OBJ_CODE   = HRB_HCD
Load or link CBDMGHOM with: (HRB,0,0,0,0)
...

```

Return codes

On return, HRB_RETURN_CODE in the request block HRB contains the severity of an error:

- HRB_SEVERE indicates that processing has been terminated and a new setup is required. Issue HCD_MGET to retrieve the messages describing the error.
- HRB_SYNTAX indicates that the request was given to the API in an incorrect syntax and therefore, the request has not been processed.
- HRB_WARNING and HRB_ERROR are given for the remaining errors. Issue HCD_MGET to retrieve the messages describing the error.
- HRB_OK tells you that no problems occurred.

Reason codes

HRB_REASON_CODE in the request block HRB specifies the error in more detail.

————— End of Programming Interface information —————

Appendix E. Establishing the host communication

To communicate with HCD from an HCM client on your workstation, you must set up the host communication depending on your operating system:

- For information regarding z/OS, refer to “Setting up TCP/IP definitions for z/OS”
- For information regarding z/VM, refer to information unit *Setting up TCP/IP definitions for z/VM in z/OS and z/VM HCM User’s Guide*.

The host communication is also used when communicating between different HCD systems, for example, for tasks like HMC-wide activation.

Setting up TCP/IP definitions for z/OS

To communicate with HCD from an HCM client or from another HCD program, a server program running on the host is required. The server program is a TCP/IP program that listens for incoming remote HCD requests on a specific TCP/IP port. These HCD requests are passed to HCD to be executed. The server program - called HCD agent - must be started before HCD requests are passed. The HCD agent (communicating with the HCD server) is started by a daemon program (HCD dispatcher) as soon as a remote HCD login request has been issued. This HCD dispatcher program must be started before the first remote HCD login request is issued.

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

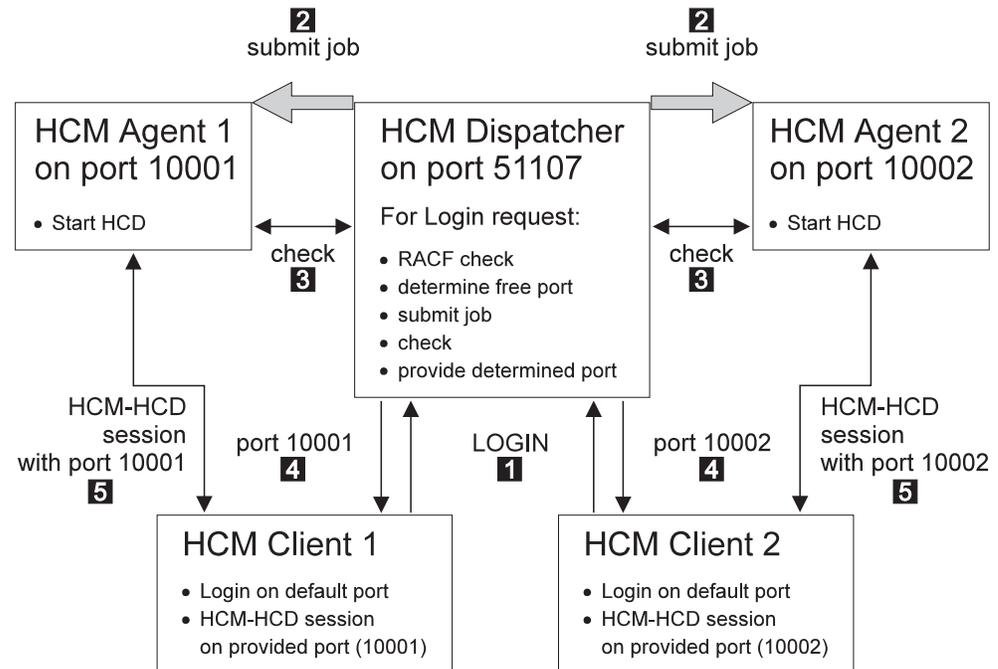


Figure 247. Relationship between HCD client, HCD dispatcher, and HCD agent

The HCD dispatcher listens on a specific TCP/IP port and waits for incoming remote HCD login requests. For each remote HCD login request **1**, the HCD dispatcher checks the passed user ID and password for correctness. If user ID and password are correct, the HCD dispatcher looks for a free IP port. Then it starts an HCD agent program (HCD server program) which listens to remote HCD requests on that particular IP port **2**. As soon as the HCD agent is started and ready **3**, the HCD dispatcher passes the particular IP port to the HCD client **4**. The HCD client then closes the session to the HCD dispatcher and starts a session to the started HCD agent using the passed IP port **5**. As soon as the HCD client has connected the HCD agent, the HCD dispatcher is free again to wait for other incoming remote HCD login requests on its IP port. As soon as HCD terminates the HCD client server connection, the HCD agent is terminated and the used port is freed again.

The advantage of having an HCD dispatcher which waits permanently on a specific port for incoming HCD login requests is, that each remote HCD 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 remote HCD users automatically have the same setup, and that this has only to be done once. As the HCD dispatcher is always running, all remote HCD users always use the same IP port for the login request.

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

The HCD dispatcher creates a job out of a skeleton and submits this job to start the HCD agent (for an example of the job skeleton, see "Skeleton used to start the HCD agent" on page 507). After the HCD agent has been started and is running, the HCD client communicates with the HCD agent. The HCD client uses the same host name for the communication with the HCD agent as it has used for the login request to the HCD dispatcher. Therefore, the HCD agent must run on a system with the same host name as the HCD 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 HCD dispatcher in two ways:

- Starting the HCD dispatcher as a started task by using the procedure CBDQDISP provided in the library SYS1.PROCLIB. Consider to start the HCD dispatcher automatically after IPL of your z/OS system (for example, by using System Automation). See "Starting the HCD dispatcher as a started task" on page 508 for an example on how to start the HCD dispatcher as a started task.
- Starting the HCD dispatcher by submitting a batch job. A sample of a job, which can be used to start the HCD 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 HCD dispatcher. Inform the remote HCD users of the TCP/IP port for the login requests. The HCD client does not accept a port number 0 or port numbers higher than 65535.

Each HCD session needs its own dedicated server, and each active HCD server needs its own unique TCP/IP port number. The HCD dispatcher looks for a free port number in a specific range. As a 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 HCD agent during start of the HCD dispatcher.

The user ID under which the HCD dispatcher is running, as well as the user IDs used for working with remote HCD programs 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 remote HCD user ID.

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

Controlling access to HCD services

To access the HCD services remotely on z/OS, a user needs to log on to the HCD dispatcher. The log-on 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 look like:

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

For details about protecting applications see *z/OS Security Server RACF Security Administrator's Guide*.

Skeleton used to start the HCD agent

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

You can adapt this skeleton according to your installation needs. You can specify accounting information in the job card of the skeleton, if your installation requires accounting information. If your installation requires accounting information, and there is no accounting information specified in the skeleton, each HCM user must provide this information in the EEQHCM.INI file. For all other substitutable parameters, the HCD dispatcher provides default values, for example, `&SYSUID..HCD.PROFILE` and `&SYSUID..HCM.TRACE` as data set names for the HCD profile and the HCD trace data set for remote HCD sessions.

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 HCD agent.

Always check, whether you must 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 HCD client logging to the HCD 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 HCD dispatcher not to use the default skeleton, but the user-specified skeleton.

Starting the HCD dispatcher as a started task

There is a procedure CBDQDISP provided in SYS1.PROCLIB, which you can use to start the HCD 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 HCD 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 HCD dispatcher always after the system has been IPLed. You can also start the HCD dispatcher automatically using System Automation for z/OS.

You can change the following parameters in the procedure:

Job name

The HCD dispatcher submits a job for each incoming remote HCD login request. As a 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 remote HCD user, specify +U and the HCD dispatcher substitutes the +U with the requestor'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 HCD dispatcher does not accept more than seven digits for the beginning of a job name. Strings longer than seven digits are truncated. The HCD dispatcher generates a job name using the JNP variable and fills it up to eight digits. If the JNP variable contains three digits, the job name is filled up to eight digits by using parts of the port address of the HCD dispatcher and parts of the port address of the HCD client, which are started. If the JNP variable does not have three digits, it is filled up to eight digits by using 0...9 and A...Z.

Logging information

For special cases (debugging or better control), the HCD 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 HCD dispatcher listens for incoming remote HCD login requests, you can specify your port by setting the PORT variable.

Skeleton

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

Port range

It is possible to determine a different range for the ports chosen by the HCD dispatcher for communication with the HCD agent during start of the dispatcher. The port range used must be configured to be available and permitted to the HCD 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 HCD 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 HCD dispatcher can also be used by the UNIX System Services. (Note that the specified home directory for the HCD dispatcher is the root directory in this example).

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

Figure 248. Sample JCL for creating a user ID for UNIX System Services

Starting the HCD dispatcher as a batch job

You can either start the HCD 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 HCD dispatcher

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

Define an HCD profile

HCM uses HCD on the host as its server. Thus you can 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 “Defining an HCD profile” on page 19.

Allocate the HCD trace data set for remote HCD sessions

Before you can start an HCD agent (HCD server) on the host, you must allocate a data set that is used by this HCD 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 remote HCD programs concurrently. The recommended name for the HCD agent trace data set is *userid.HCD.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.HCD.TRACE,
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=6160),
// SPACE=(CYL,(5,20)), DISP=(NEW,CATLG), UNIT=SYSALLDA
//
```

Figure 249. Sample job for trace data set allocation

Verifying TCP/IP host communication

If you cannot establish a connection to the host, first check whether you specified the correct host name in the remote HCD 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 remote HCD host name.

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

```
ping <nameserver>
```

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

TCP/IP problem determination

If a user connects to HCD via HCM, HCM displays error messages if the TCP/IP connection fails. For information about error messages, refer to *z/OS and z/VM HCM User's Guide*. You can also check for documentation updates in *HCM Documentation* under *Product Updates* provided on the HCM home page (<http://www.ibm.com/systems/z/os/zos/features/hcm/>).

If you connect to HCD via another HCD instance, HCD displays an error message on the screen or in the HCD message log.

If you get messages that are not listed in *z/OS and z/VM HCM User's Guide* or in *z/OS and z/VM HCD Messages*, 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.

Appendix F. IODF data model

The following tables describe the IODF data model in terms of object class and attribute definitions. These definitions conform to the data that is kept in the IODF.

The object classes and attribute types introduced by HCD start with the prefix **hcd** which helps with identifying them.

The object hierarchy preceding the HCD Backend suffix is structured as follows:

hcdIodf (RDN: hcdIodfId)



IODF object class definitions within LDAP

The following tables describe the object classes which are used for the HCD LDAP DIT. In addition to other information, they contain the names of the attributes of each object class. The required attributes (shown in the row 'Must Contain') must be specified when adding an object of that class. Their values cannot be deleted. Optional attributes (shown in the row 'May Contain') may not always be required. They may, however, be necessary for certain objects according to HCD configuration rules. Attributes which are used to build the RDN of an object must not be modified for the corresponding object class.

The row 'Supported Requests' contains information about which operations are possible on entries of the corresponding object class. There are some object classes which do not allow all contained (non-RDN) attributes to be added (because they are read-only attributes) or modified. In this case, the list of attributes which may be modified is specified in the 'Supported Requests' row in parentheses after the MODIFY identifier.

Note: For certain entries, there may be restrictions to this general description as a result of HCD configuration rules.

The following rules apply for update requests on control unit and device object classes:

1. Control Unit

The general attributes of a specific control unit exist in an appropriate entry in class `hcdControlUnit`. For each connected processor, the processor related attributes of that control unit are contained in an entry of class `hcdCssControlUnit`. Hence, as a result of this relation, an entry can only be added to class `hcdCssControlUnit` if the corresponding entry in class `hcdControlUnit` is already present. Likewise, if an entry of class `hcdControlUnit` is deleted, all the entries for the corresponding control unit in class `hcdCssControlUnit` will implicitly also be deleted.

2. Devices

Similar to the control units, the general attributes of a specific device exist in an entry of class `hcdDevice` and may additionally exist in entries of classes `hcdCssDevice` and `hcdOsDevice`. The device attributes relating a device to a particular processor are contained in a corresponding entry of class `hcdCssDevice`. Such an entry can only exist if an entry for the device in class `hcdDevice` also exists and the device is connected to a control unit defined via, both an entry in the class `hcdControlUnit`, and an entry in class `hcdCssControlUnit`. This dependency means that it is not possible to explicitly add or delete an entry of class `hcdCssDevice` directly. Instead, an entry of class `hcdCssDevice` will be implicitly added if:

- A control unit defined in an `hcdControlUnit` entry that has the particular device attached to it is defined also as an entry in class `hcdCssControlUnit`.
- An entry in class `hcdDevice` is created containing the attribute `hcdConnControlUnits` with a control unit number of an `hcdControlUnit` entry that also has a corresponding `hcdCssControlUnit` entry defined.

Implicitly added entries of class `hcdCssDevice` initially have default attribute values. These attributes can be modified to get their proper values.

An entry of class `hcdCssDevice` will be implicitly deleted if:

- The entry in corresponding class hcdDevice is deleted, or
- The connected control units in class hcdCssControlUnit are deleted, or
- The connected control units in class hcdControlUnit are deleted.

Changes to the attributes of control unit and device entries may affect all corresponding entries in the same logical control unit of a processor configuration, i.e. all entries of classes hcdCssControlUnit and hcdCssDevice with the same value of the attribute hcdLogicalControlUnit. In such a case, it may be necessary to delete all entries belonging to the same logical control unit and add them again with the changed attributes. This should be done using the update sequence in transaction mode.

For each operating system configuration, the operating system related device attributes are contained in a corresponding entry in class hcdOsDevice. An entry in class hcdOsDevice can only exist if the corresponding entry in class hcdDevice also exists. An entry in class hcdOsDevice can be explicitly added or deleted. This means that the device is connected to or disconnected from the operating system configuration. An entry in class hcdOsDevice is implicitly deleted if the corresponding entry in class hcdDevice is deleted.

If a device entry in class hcdDevice is added which has the same RDN attribute (hcdDeviceNumber) value of an existing entry in that class, HCD assigns a non-zero suffix to the device number (RDN attribute hcdDeviceSuffix) to make the RDN unique. The attribute hcdDeviceSuffix cannot be specified with an add request. If the value of attribute hcdDeviceSuffix is 0000, it need not be specified in the RDN, as this is the default value. If the value of the attribute hcdDeviceSuffix is not equal to 0000, it has to be specified in the RDN of the MODIFY and DELETE requests for the corresponding device entry. Therefore, the complete device RDN has to be retrieved via a preceding search request in order to be able to specify the correct RDN of a device entry.

Note: To facilitate the search of the correct hcdDeviceSuffix of devices in a device group, you can specify an LDAP search request for a specific device number, but without device suffix, or with device suffix 0000. Such a request will return all device entries with the same device number with its different device suffixes. From the returned entries, you can retrieve the suffix for the device that you now can modify or delete with a subsequent request.

Class	hcdChannelPath
Description	Describes a channel path of the processor configuration
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdProcessorConfig
Must Contain	hcdChannelPathId, hcdChannelPathType, hcdOperationMode, objectClass
May Contain	hcdDescription, hcdDynSwitch, hcdConnChannelPath, hcdAccessingPartitions, hcdCandidatePartitions, hcdConnPort, hcdIsOccupied, hcdIsManaged, hcdSysplex, hcdMaximumFrameSize, hcdSpanningChannelSubsystems, hcdPhysicalChannelId, hcdHasPrioQueuesDisabled, hcdHcaAdapterId, hcdHcaPort, hcdChannelFunction, hcdPhysicalNetworkIds, hcdVirtualChannelId

Class	hcdChannelPath
Supported Requests	SEARCH ADD MODIFY DELETE
Special Notes	Attribute hcdIsOccupied is only shown if set to Yes. Attribute hcdSpanningChannelSubsystems is output only. A spanned channel path is added/modified in one channel subsystem while specifying partitions from more than one channel subsystem. It can be defined with operation mode SHR or SPAN. Attribute hcdHasPrioQueuesDisabled is only shown if applicable.

Class	hcdControlUnit
Description	Describes a control unit
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdIodf
Must Contain	hcdControlUnitNumber, hcdUnit, objectClass
May Contain	hcdDescription, hcdModel, hcdSerialNumber, hcdConnPorts
Supported Requests	SEARCH ADD MODIFY DELETE

Class	hcdCssControlUnit
Description	Describes a control unit from the Channel Subsystem viewpoint
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdProcessorConfig
Must Contain	hcdControlUnitNumber, hcdConnChannelPaths, objectClass
May Contain	hcdUnitAddressRanges, hcdControlUnitAddress, hcdIOConcurrencyLevel, hcdControlUnitProtocol, hcdLogicalControlUnit, hcdManagedChannelPathsCount
Supported Requests	SEARCH ADD (requires the existence of an object in class hcdControlUnit with same RDN) MODIFY (hcdConnChannelPaths, hcdUnitAddressRanges, hcdControlUnitAddress, hcdIOConcurrencyLevel, hcdControlUnitProtocol, hcdManagedChannelPathsCount) DELETE

Class	hcdCssDevice
Description	Describes a device from the Channel Subsystem viewpoint
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdProcessorConfig
Must Contain	hcdDeviceNumber, objectClass
May Contain	hcdDeviceSuffix, hcdDeviceRange, hcdUnitAddress, hcdStatusDetection, hcdTimeOut, hcdPreferredChannelPath, hcdCandidatePartitions, hcdLogicalControlUnit, hcdSubchannelSetId
Supported Requests	SEARCH MODIFY (hcdDeviceRange, hcdUnitAddress, hcdStatusDetection, hcdTimeOut, hcdPreferredChannelPath, hcdCandidatePartitions)
Special Notes	For a MODIFY request, hcdDeviceRange specifies the number of devices to which the request will be applied.

Class	hcdDevice
Description	Describes an I/O device
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdIodf
Must Contain	hcdDeviceNumber, hcdUnit, objectClass
May Contain	hcdDescription, hcdModel, hcdDeviceSuffix, hcdDeviceRange, hcdSerialNumber, hcdVolser, hcdConnControlUnits, hcdDevicePprcUsage
Supported Requests	SEARCH ADD MODIFY (hcdDeviceRange, hcdUnit, hcdDescription, hcdModel, hcdSerialNumber, hcdVolser, hcdConnControlUnits) DELETE
Special Notes	For ADD and MODIFY requests, hcdDeviceRange specifies the number of devices to which the request will be applied.

Class	hcdDistPackage
Description	Describes a distribution package of I/O configurations
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdIodf
Must Contain	hcdDistPackageId, objectClass

Class	hcdDistPackage
May Contain	hcdDescription, hcdTargetNode, hcdTargetUser, hcdIsAttended, hcdTargetIodf, hcdTargetVolser, hcdSentDate, hcdSentTime, hcdOsConfigs, hcdProcessorConfigs
Supported Requests	SEARCH ADD MODIFY (hcdDescription, hcdTargetNode, hcdTargetUser, hcdIsAttended, hcdTargetIodf, hcdTargetVolser, hcdOsConfigs, hcdProcessorConfigs) DELETE

Class	hcdEligibleDeviceTable
Description	Describes an Eligible Device Table of an OS configuration
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdOsConfig
Must Contain	hcdEligibleDeviceTableId, objectClass
May Contain	hcdDescription, hcdLastUpdateDate, hcdUpdatedBy
Supported Requests	SEARCH ADD MODIFY (hcdDescription) DELETE

Class	hcdEsotericDeviceGroup
Description	Describes an esoteric device group of an EDT of an OS configuration
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdEligibleDeviceTable
Must Contain	hcdEsotericDeviceGroupId, objectClass
May Contain	hcdEsotericDeviceGroupToken, hcdVirtualIO, hcdDeviceRanges
Supported Requests	SEARCH ADD MODIFY DELETE
Special Notes	Attribute hcdVirtualIO is only shown if set to Yes.

Class	hcdGenericDeviceType
Description	Describes an update to a generic device type for an EDT of an OS configuration
Type	structural

Class	hcdGenericDeviceType
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdEligibleDeviceTable
Must Contain	hcdGenericDeviceTypeId, objectClass
May Contain	hcdDevicePreferenceValue, hcdVirtualIO
Supported Requests	SEARCH MODIFY
Special Notes	<p>Entries of class hcdGenericDeviceType are implicitly added when the first device entry with the corresponding generic device type is added in class hcdOsDevice.</p> <p>An entry of this class is deleted, if the last device with the corresponding generic device type is deleted from class hcdOsDevice.</p> <p>Attribute hcdVirtualIO is only shown if set to Yes.</p>

Class	hcdIocds
Description	Describes an IOCDS of a processor configuration
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdProcessorConfig
Must Contain	hcdIocdsId, objectClass
May Contain	hcdIocdsName, hcdLastUpdateDate, hcdLastUpdateTime, hcdProcessorConfigMode
Supported Requests	SEARCH
Special Notes	<p>Entries in this class are implicitly added when an entry in class hcdProcessorConfig is added. Entries in this class are implicitly deleted if the appropriate entry in class hcdProcessorConfig is deleted.</p>

Class	hcdIodf
Description	Describes the I/O configurations defined in an IODF
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	
Must Contain	hcdIodfId, objectClass
May Contain	hcdIodfType, hcdIodfDescription, hcdBackupIodf, hcdBlocksAllocated, hcdBlocksUsed, hcdCreationDate, hcdLastUpdateDate, hcdLastUpdateTime, hcdToken (output only)
Supported Requests	SEARCH MODIFY (hcdIodfDescription)

Class	hcdIodf
Special Notes	Entries of this class cannot be added or deleted with the HCD LDAP Backend.

Class	hcdOsConfig
Description	Describes an OS configuration
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdIodf
Must Contain	hcdOSConfigId, hcdOSConfigType, objectClass
May Contain	hcdDescription, hcdOSDrSiteConfigId
Supported Requests	SEARCH ADD MODIFY (hcdDescription) DELETE

Class	hcdOsDevice
Description	Describes an I/O device from the OS viewpoint
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdOsConfig
Must Contain	hcdDeviceNumber, hcdUnit, objectClass
May Contain	hcdDeviceSuffix, hcdDeviceRange, hcdModel, hcdGenericDeviceTypeId, hcdDeviceParameters, hcdDeviceFeatures, hcdConsoleNumber, hcdSubchannelSetId
Supported Requests	SEARCH ADD (requires the existence of an object in class hcdDevice with the same RDN) MODIFY (hcdDeviceRange, hcdDeviceParameters, hcdDeviceFeatures, hcdConsoleNumber) DELETE
Special Notes	For ADD and MODIFY requests, hcdDeviceRange specifies the number of devices to which the request will be applied.

Class	hcdPartition
Description	Describes a logical partition (LP) of a processor configuration
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdProcessorConfig
Must Contain	hcdPartitionId, objectClass

Class	hcdPartition
May Contain	hcdDescription, hcdPartitionNumber, hcdPartitionUsage, hcdUniqueId
Supported Requests	SEARCH ADD MODIFY DELETE
Special Notes	You can use the hcdPartitionNumber attribute to identify a reserved partition.

Class	hcdPCIeFunction
Description	Describes a PCIe function of a processor configuration
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdProcessorConfig
Must Contain	hcdPCIeFunctionId, objectClass,
May Contain	hcdPCIeFunctionType, hcdDescription, hcdAccessingPartitions, hcdCandidatePartitions, hcdPhysicalChannelId, hcdVirtualFunctionNum, hcdPhysicalNetworkIds, hcdVirtualChannelId, hcdPCIeUID
Supported Requests	SEARCH ADD MODIFY DELETE

Class	hcdPort
Description	Describes a port of an ESCON director
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdSwitch
Must Contain	hcdPortId, objectClass
May Contain	hcdPortName, hcdIsOccupied, hcdIsInstalled, hcdConnPort
Supported Requests	SEARCH MODIFY
Special Notes	Attribute hcdIsOccupied is only shown if set to Yes. Entries of this class are implicitly added or deleted if the corresponding switch entry in class hcdSwitch is added or deleted, respectively.

Class	hcdPortConfig
Description	Describes a port configuration of a switch configuration
Type	structural

Class	hcdPortConfig
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdSwitchConfig
Must Contain	hcdPortId, objectClass
May Contain	hcdAllowedDynConnPorts, hcdProhibitedDynConnPorts, hcdDedicatedPort, hcdIsBlocked
Supported Requests	SEARCH MODIFY
Special Notes	Entries in this class are implicitly added or deleted if the corresponding entry in class hcdSwitchConfig is added or deleted, respectively.

Class	hcdProcessorConfig
Description	Describes a processor configuration or, if it is identified by an RDN of the form hcdProcessorConfigId+CssId, it describes a channel subsystem of an XMP processor.
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdIodf
Must Contain	hcdProcessorConfigId, hcdUnit, hcdProcessorConfigMode, objectClass
May Contain	hcdModel, hcdDescription, hcdSerialNumber, hcdNetworkName, hcdSystem, hcdSupportLevel, hcdCssId, hcdSetZeroMaxDevices, hcdSetOneMaxDevices, hcdSetTwoMaxDevices, hcdSetThreeMaxDevices, hcdLocalSystemName, hcdToken (output only)
Supported Requests	SEARCH ADD MODIFY DELETE
Special Notes	Adding a channel subsystem requires that the processor configuration has already been added. To add, change or delete a channel subsystem, an RDN consisting of hcdProcessorConfigId+hcdCssId has to be used. The only applicable further attributes for a channel subsystem are <i>hcdDescription</i> , <i>hcdSetZeroMaxDevices</i> , <i>hcdSetOneMaxDevices</i> , and <i>hcdSetTwoMaxDevices</i> . The values of the required parameters <i>hcdUnit</i> and <i>hcdProcessorConfigMode</i> are ignored.

Class	hcdSwitch
Description	Describes a switch (ESCON or FICON Director or Fibre Channel switch)
Type	structural
Derived from	top
Auxiliary Classes	

Class	hcdSwitch
Possible Superiors	hcdIodf
Must Contain	hcdSwitchId, hcdUnit, objectClass
May Contain	hcdDescription, hcdModel, hcdSerialNumber, hcdSwitchAddress
Supported Requests	SEARCH ADD MODIFY DELETE

Class	hcdSwitchConfig
Description	Describes a configuration of an ESCON Director
Type	structural
Derived from	top
Auxiliary Classes	
Possible Superiors	hcdSwitch
Must Contain	hcdSwitchConfigId, objectClass
May Contain	hcdDescription, hcdDefaultConn
Supported Requests	SEARCH ADD MODIFY DELETE

IODF attribute table

The following table describes the names and the properties of the attributes. The first column additionally shows the alias names in parentheses where available.

Note:

1. The *Access Class* for all attributes is *normal*.
2. All attribute type names ending with the plural *s* represent multi-valued attributes. HCD interprets the values as an ordered list. This is important if there are two lists which have corresponding list items.
3. In a request, a mix of base names and alias names for attributes is not supported.

Attribute (alias name)	Description	Syntax	Valued
hcdAccessingPartitions (hcdAccList)	Channel path access list (up to 8 char partition names)	cis / 8	multi
hcdAllowedDynConnPorts (hcdAllowedList)	Ports to which an allowed dynamic connection exists (2 hexadecimal digit port address)	cis / 2	multi

Attribute (alias name)	Description	Syntax	Valued
hcdBackupIodf (hcdBkupIodf)	Data set name of backup IODF (up to 35 character full-qualified IODF data set name)	cis / 35	single
hcdBlocksAllocated (hcdAllocBlks)	Number of allocated blocks for IODF (up to 6 digit decimal number, read-only)	cis / 6	single
hcdBlocksUsed (hcdUsedBlks)	Number of used blocks in IODF (up to 6 digit decimal number, read-only)	cis / 6	single
hcdCandidatePartitions (hcdCandList)	Partition candidate list (up to 8 character partition names or a 0 (zero) for a null device candidate list)	cis / 8	multi
hcdChannelFunction	Channel function: blank indicates basic HiperSockets support X indicates IEDN Access B indicates External Bridge support	cis / 1	single
hcdChannelPathId (hcdChpId)	Channel path identifier (2 digit hexadecimal number)	cis / 2	single
hcdChannelPathType (hcdChpType)	Channel path type (4 character channel path type acronym)	cis / 4	single
hcdConnChannelPath (hcdConnChp)	<p>Connected channel path of a coupling facility connection (qualified coupling facility connection: prid.chpid.cunum.devn.cnt (if the target is an SMP processor) or prid.cssid.chpid.cunum.devn.cnt (if the target is an XMP processor), where</p> <ul style="list-style-type: none"> prid is a up to 8 character processor name, cssid is a one hexadecimal digit channel subsystem ID, chpid is a 2 hexadecimal digit channel path, cunum is a 4 hexadecimal digit control unit number, devn is a 4 hexadecimal digit device number, cnt is an up to 3 decimal device count (without leading zeros) <p>If the values for cunum, devn or cnt are missing, they are defaulted as needed.</p>	cis / 27	single

Attribute (alias name)	Description	Syntax	Valued
hcdConnChannelPaths (hcdChpList)	Connected channel paths / link addresses (pp or pp.ll or pp.llll, where pp is the 2-digit hexadecimal number of the channel path, optionally qualified by ll as a 2 hexadecimal number for the link address or by llll as a 4 hexadecimal number for the link address)	cis / 7	multi
hcdConnControlUnits (hcdCuList)	Connected control units (4 hex digit numbers)	cis / 4	multi
hcdConnPort	Connected switch port (qualified port address ss.pp, where ss is the 2 digit hexadecimal switch number, pp is the 2 digit hexadecimal port address)	cis / 5	single
hcdConnPorts (hcdConnPortList)	List of connected switch ports (qualified port addresses ss.pp, where ss is the 2 digit hexadecimal switch number, pp is the 2 digit hexadecimal port address)	cis / 5	multi
hcdConsoleNumber (hcdConsole)	Console order number (up to 2 digit decimal number)	cis / 2	single
hcdControlUnitAddress (hcdCuadd)	Logical control unit address (up to 2 hex digit number)	cis / 2	single
hcdControlUnitNumber (hcdCu)	Control unit number (4 hex digit number)	cis / 4	single
hcdControlUnitProtocol (hcdProt)	Control unit protocol (D, S, S4)	cis / 2	single
hcdCreationDate	Creation Date (yyyy-mm-dd, read-only)	cis / 10	single
hcdCssId	Channel subsystem identifier (1 digit hexadecimal number)	cis / 1	single
hcdDedicatedPort (hcdDedicated)	Port to which a dedicated connection exists (2 digit hexadecimal number)	cis / 2	single
hcdDefaultConn	Dynamic connection default of a switch configuration (ALLOW, PROHIBIT)	cis / 10	single
hcdDescription	Description field (up to 32 character string)	cis / 32	single

Attribute (alias name)	Description	Syntax	Valued
hcdDeviceFeatures (hcdFeatList)	Device feature list (up to 10 character identifier)	cis / 10	multi
hcdDeviceNumber (hcdDev)	Device number (4 digit hexadecimal number)	cis / 4	single
hcdDeviceParameters (hcdParmList)	Device parameter list (keyword=value1,...)	cis / 50	multi
hcdDevicePprcUsage	Device PPRC usage (D, F, S, U, or N)	cis / 1	single
hcdDevicePreferenceValue (hcdPrefVal)	Device preference value (up to 8 digit decimal number)	cis / 8	single
hcdDeviceRange (hcdRange)	Range of devices with same attributes (up to 4 digit decimal number)	cis / 4	single
hcdDeviceRanges (hcdDevRngeList)	Device range list (qualified value: xxxx.ddd, where xxxx is a 4 hex digit device number, ddd is a up to 3 decimal digit range number,	cis / 8	multi
hcdDeviceSuffix (hcdSuf)	Device ID suffix (4 digit hexadecimal number)	cis / 4	single
hcdDistPackageId (hcdPkgName)	Name of distribution package (up to 8 alphanumeric character identifier)	cis / 8	single
hcdDynSwitch	Dynamic switch of channel path (2 digit hexadecimal number)	cis / 2	single
hcdEligibleDeviceTableId (hcdEdtId)	EDT identifier (2 character identifier)	cis / 2	single
hcdEsotericDeviceGroupId (hcdEsoId)	Esoteric name (up to 8 character identifier)	cis / 8	single
hcdEsotericDeviceGroupToken (hcdEsoToken)	Esoteric token (up to 4 decimal digit number)	cis / 4	single

Attribute (alias name)	Description	Syntax	Valued
hcdGenericDeviceTypeId (hcdGenId)	Generic device type name (up to 8 character identifier)	cis / 8	single
hcdHasPrioQueuesDisabled	Indicates if an OSD, OSM, or OSX channel path has queue prioritizing disabled (Yes, No).	cis / 3	single
hcdHcaAdapterId	HCA adapter ID (hexadecimal number or * for over-defined channel path).	cis / 2	single
hcdHcaPort	HCA port (decimal number).	cis / 1	single
hcdIocdsId	IOCDS identifier (2 digit hexadecimal number)	cis / 2	single
hcdIocdsName	IOCDS name (up to 8 character identifier)	cis / 8	single
hcdIOConcurrencyLevel (hcdIoCl)	I/O concurrency level of control unit (1 digit decimal number)	cis / 1	single
hcdIodfDescription (hcdIodfDesc)	IODF description (up to 128 character string)	cis / 128	single
hcdIodfId	IODF name (up to 35 character full-qualified data set name)	cis / 35	single
hcdIodfType	IODF type (I (initial), W (work), P (production), read-only)	cis / 1	single
hcdIsAttended	Indicates if target node is attended (Yes, No)	cis / 3	single
hcdIsBlocked	Indicates if the port is blocked (Yes, No)	cis / 3	single
hcdIsInstalled	Indicates if object (port) is installed (Yes, No)	cis / 3	single
hcdIsManaged	Indicates if a channel path is managed (Yes, No)	cis / 3	single
hcdIsOccupied (hcdIsOcc)	Indicates if object (port, channel path) is occupied (Yes, No)	cis / 3	single
hcdLastUpdateDate (hcdLastUpdDate)	Date of last update (yyyy-mm-dd, read-only)	cis / 10	single
hcdLastUpdateTime (hcdLastUpdTime)	Time of last update (hh:mm:ss, read-only)	cis / 8	single

Attribute (alias name)	Description	Syntax	Valued
hcdLocalSystemName	Local system name (up to 8 alphanumeric character identifier)	cis / 8	single
hcdLogicalControlUnit (hcdLcu)	Logical control unit number (4 digit hexadecimal number, read-only)	cis / 4	single
hcdManagedChannelPathsCount	Maximum number of managed channel path connections to a control unit (1 digit decimal number)	cis / 1	single
hcdMaximumFrameSize	Maximum frame size for the transmission unit on an IQD channel path in KB (decimal number)	cis / 6	single
hcdModel	Model (4 character identifier)	cis / 4	single
hcdNetworkName (hcdNetName)	CPC network name (up to 8 alphanumeric identifier)	cis / 8	single
hcdOperationMode (hcdOpMode)	Channel path operation mode (DED, REC, SHR)	cis / 3	single
hcdOSConfigId 922 (hcdOSId)	OS configuration identifier (up to 8 alphanumeric character identifier)	cis / 8	single
hcdOSDrSiteConfigId	D/R (disaster recovery) site OS configuration identifier (up to 8 alphanumeric character identifier)	cis / 8	single
hcdOSConfigs (hcdOSList)	List of OS configurations (up to 8 alphanumeric character identifiers)	cis / 8	multi
hcdOSConfigType (hcdOSType)	OS configuration type (MVS, VM)	cis / 3	single
hcdPartitionId (hcdPartId)	Logical partition name (up to 8 alphanumeric character identifier or an * for reserved partitions)	cis / 8	single
hcdPartitionNumber (hcdPartNo)	Partition image number (1 digit hexadecimal number)	cis / 1	single

Attribute (alias name)	Description	Syntax	Valued
hcdPartitionUsage (hcdPartUsage)	Partition usage (OS, CF, CF/OS)	cis / 5	single
hcdPCIeFunctionId	PCIe function identifier (4 digit hexadecimal number)	cis / 4	single
hcdPCIeFunctionType	PCIe function type identifier (12 char name)	cis / 12	single
hcdPCIeUID	PCIe unique identifier (8 digit hexadecimal number)	cis / 8	single
hcdPhysicalChannelId	Physical channel identifier for a channel path of an XMP processor (3 digit hexadecimal number or '*' for an over-defined channel path)	cis / 3	single
hcdPhysicalNetworkIds	Physical network ID list (up to 16 char name each)	cis / 16	multi
hcdPortId	Switch port identifier (2 digit hexadecimal number)	cis / 2	single
hcdPortName	Switch port name (up to 24 character identifier)	cis / 24	single
hcdPreferredChannelPath (hcdPrefChp)	Preferred channel path (2 digit hexadecimal number)	cis / 2	single
hcdProcessorConfigId (hcdProcId)	processor configuration ID (up to 8 alphanumeric character identifier)	cis / 8	single
hcdProcessorConfigMode (hcdConfMode)	Processor configuration mode (BASIC, LPAR)	cis / 5	single
hcdProcessorConfigs (hcdProcList)	List of processor configurations (up to 8 alphanumeric character identifiers)	cis / 8	multi
hcdProhibitedDynConnPorts (hcdProhibitList)	Ports to which a Prohibited dynamic connection exists (2 digit hexadecimal numbers)	cis / 2	multi
hcdSentDate	Date of last distribution (yyyy-mm-dd, read-only)	cis / 10	single
hcdSentTime	Time of last distribution (hh:mm:ss, read-only)	cis / 8	single
hcdSerialNumber (hcdSerialNo)	Serial number (up to 10 character serial number)	cis / 10	single

Attribute (alias name)	Description	Syntax	Valued
hcdSetOneMaxDevices	Maximum number of devices for subchannel set 1 (SS 1) in a channel subsystem (6 digit decimal number).	cis / 6	single
hcdSetThreeMaxDevices	Max number of devices for subchannel set 3 in a channel subsystem (6 digit decimal number)	cis / 6	single
hcdSetTwoMaxDevices	Maximum number of devices for subchannel set 2 (SS 2) in a channel subsystem (6 digit decimal number).	cis / 6	single
hcdSetZeroMaxDevices	Maximum number of devices for subchannel set 0 (SS 0) in a channel subsystem of an XMP processor (6 digit decimal number).	cis / 6	single
hcdSpanningChannelSubsystems	List of spanning channel subsystems for a spanned channel path (1 digit hexadecimal number, read only).	cis / 1	multi
hcdStatusDetection (hcdStadet)	Status detection facility of device (Yes, No)	cis / 3	single
hcdSubchannelSetId	ID of the subchannel set where a device is located in a channel subsystem of an XMP processor.	cis / 1	single
hcdSupportLevel	Processor support level (8 character identifier)	cis / 8	single
hcdSwitchAddress	Address of a Fibre Channel switch in a fabric (2 digit hexadecimal number)	cis / 2	single
hcdSwitchConfigId (hcdSwConfId)	Switch configuration identifier (up to 8 alphanumeric character identifier)	cis / 8	single
hcdSwitchId (hcdSwID)	Switch identifier (2 digit hexadecimal number)	cis / 2	single
hcdSysplex	Sysplex name (up to 8 character alphanumeric identifier)	cis / 8	single
hcdSystem	CPC system name (up to 8 character alphanumeric identifier)	cis / 8	single
hcdTargetIodf	Data set name of target IODF (up to 35 character full-qualified data set name)	cis / 35	single
hcdTargetNode	Target node (up to 8 character alphanumeric identifier)	cis / 8	single
hcdTargetUser	Target user (up to 8 character alphanumeric identifier)	cis / 8	single
hcdTargetVolser (hcdTargetVol)	Volume serial number of target IODF data set (up to 6 character alphanumeric identifier)	cis / 6	single

Attribute (alias name)	Description	Syntax	Valued
hcdTimeOut	Time out facility (Yes, No)	cis / 3	single
hcdToken	World wide unique token	cis / 64	
hcdUnit	Unit (up to 8 alphanumeric character identifier)	cis / 8	single
hcdUnitAddress (hcdUa)	Unit address (2 digit hexadecimal number)	cis / 2	single
hcdUnitAddressRanges (hcdUaList)	Unit address range of control unit (qualified number xx.ddd, where xx is a 2 digit hexadecimal unit address ddd is a up to 3 digit decimal range)	cis / 6	multi
hcdUniqueUid	uniqueness flag	cis / 1	single
hcdUpdatedBy	Identifier of last update user (up to 8 alphanumeric character identifier, read-only)	cis / 8	single
hcdVirtualChannelId	Virtual channel Identifier (3 digit hexadecimal number)	cis / 3	single
hcdVirtualFunctionNum	Virtual function number (3 digit decimal number)	cis / 3	single
hcdVirtualIO (hcdVio)	Virtual I/O (Yes, No)	cis / 3	single
hcdVolser (hcdVolume)	Volume serial number (up to 6 character alphanumeric identifier)	cis / 6	single

Appendix G. Accessibility

Accessible publications for this product are offered through IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter/SSLTBW/welcome>).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the <http://www.ibm.com/systems/z/os/zos/webqs.html> 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 V2R2 ISPF User's Guide Vol I*

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM Knowledge Center 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) 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 offered in the U.S.A. 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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

COPYRIGHT LICENSE:

This information might contain 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.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS™, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted

for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

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: <http://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 at <http://www.ibm.com/legal/copytrade.shtml>.

Programming interface information

This guide primarily documents information that is NOT intended to be used as a Programming Interface of Hardware Configuration Definition (HCD).

This information unit also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of HCD. This information is identified where it occurs, either by an introductory statement to a topic or by the following marking:

┌────────────────── **Programming Interface information** ───────────────────┐

└────────────────── **End of Programming Interface information** ───────────────────┘

Glossary

This glossary defines technical terms and abbreviations used in the Hardware Configuration Definition (HCD) documentation. If you do not find the term you are looking for, refer to the index of the appropriate HCD document or view the *IBM Glossary of Computing Terms*, available from: <http://www.ibm.com/ibm/terminology>.

Activity log

The activity log is a sequential data set with the name of the associated **IODF** and the suffix **ACTLOG**. Use the activity log to document all definitions you made to the current IODF using HCD.

Base Base is the base device number of a **multiple exposure** device, which is accessible by more than one device number. You assign the first device number and the system generates the additional device numbers.

Central processor complex (CPC)

A physical collection of hardware that consists of central storage, one or more central processors, timers, and channels.

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.

Change log

The change log is a VSAM data set with the name of the associated **IODF** and the suffix **CHLOG**. It will be automatically created if change logging and automatic activity logging is active. A subset of its generated entries will then be used to create the activity log entries.

Channel subsystem (CSS)

A collection of subchannels that directs the flow of information between I/O devices and main storage. It uses one or more channel paths as the communication link in managing the flow of information to or from I/O devices. Within the CSS is

one subchannel set and logical partitions. One subchannel from the set is provided for and dedicated to each I/O device accessible to the CSS. Logical partitions use subchannels to communicate with I/O devices. The maximum number of CSSs supported by a processor also 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 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.

CIB Coupling over InfiniBand. A channel path type to exploit InfiniBand technology for coupling connections.

CMT See *CHPID Mapping Tool*.

Coupling Facility (CF)

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 processor complexes, special types of high-speed, CF channels are required to provide the connectivity. A receiving CF channel path, attached to a CF partition, is to be connected to a sending CF channel path, attached to a partition in which an operating system (OS) is running.

Coupling Facility Channel

A high bandwidth fiber optic channel that provides the high-speed connectivity required for data sharing between a coupling facility and the central processor complexes directly attached to it.

CSS See *channel subsystem*.

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.

D/R site OS configuration

HCD can automatically generate a 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.

Dynamic reconfiguration

The ability to make changes to the channel subsystem and to the operating system while the system is running.

EDT An EDT (eligible device table) is an installation-defined and named representation of the devices that are eligible for allocation. The EDT defines the **esoteric** and **generic** relationship of these devices. During IPL, the installation identifies the EDT that the operating system uses. After IPL, jobs can request device allocation from any of the esoteric device groups assigned to the selected EDT. An EDT is identified by a unique ID (two digits), and contains one or more esoterics and generics. Define at least one EDT for each operating system configuration.

Enterprise Systems Connection (ESCON)

A set of products and services that provides a dynamically connected environment using optical cables as a transmission medium.

ESCON Manager (ESCM)

A licensed program that provided host control to help manage connections that use ESCON Directors. The functionality has been incorporated into the I/O

Operations component of System Automation for z/OS.

ESCON Multiple Image Facility(EMIF)

EMIF is now referred to as MIF. See below.

Esoteric

Esoteric (or esoteric device group) is an installation-defined and named grouping of I/O devices of usually the same device group. EDTs define the esoteric and generic relationship of these devices. The name you assign to an esoteric is used in the JCL DD statement. The job then allocates a device from that group instead of a specific device number or **generic** device group.

FICON

Fiber Connection Environment (FICON) is an improved optical fiber communication method offering channels with high data rate, high bandwidth, increased distance and a greater number of devices per control unit for S/390 systems. It can work together with, or replace ESCON links.

Generic

Generic (or generic device type) is an MVS-defined grouping of devices with similar characteristics. For example: the device types 3270-X, 3277-2, 3278-2, -2A, -3, -4, and 3279-2a, -2b, -2c, -3a, -3b belong to the same generic. Every generic has a generic name that is used for device allocation in the JCL DD statement. MVS interprets this name as "take any device in that group". In an operating system configuration, each EDT has the same list of generics. This list can only vary by the **preference** values and **VIO** indicators that are assigned to the generics.

HCA With Coupling over InfiniBand, a peer coupling link is emulated on a Host Communication Adapter (HCA). See also *CIB*.

HCPRIO data set

The data set containing a real I/O configuration of a VM system.

Hardware Management Console

A console used to monitor and control hardware such as the systems of a CPC.

IOCDs

An input/output configuration data set

(IOCDS) contains different configuration definitions for the selected processor. Only one IOCDS is used at a time. The IOCDS contains I/O configuration data on the files associated with the processor controller on the host processor, as it is used by the channel subsystem. The CSS uses the configuration data to control I/O requests. The IOCDS is built from the production IODF.

I/O Cluster

An I/O cluster is a sysplex that owns a managed channel path for an LPAR processor configuration.

IOCP An IOCP (I/O configuration program) is the hardware utility that defines the hardware I/O configuration to the channel subsystem. For this definition IOCP retrieves information from the IOCP input data set about the following: the channel paths in the processor complex, control units attached to the channel paths, and I/O devices assigned to the control unit. HCD users can build the IOCP input data set from a production IODF.

IODF An IODF (input/output definition file) is a VSAM linear data set that contains I/O definition information. This information includes processor I/O definitions (formerly specified by **IOCP** input streams) and operating system I/O definitions (formerly specified by **MVSCP** input streams). A single IODF can contain several processor and several operating system I/O definitions. See also "Master IODF."

I/O Operations

A component of System Automation for z/OS providing functionality formerly available with ESCON Manager.

Initial program load (IPL)

The process that loads the system programs from the auxiliary storage, checks the system hardware, and prepares the system for user operations.

LCSS Logical channel subsystems. See also *channel subsystem*.

LDAP LDAP (lightweight directory access protocol) is an Internet protocol standard, based on the TCP/IP protocol and serves to access and manipulate data organized

in a Directory Information Tree (DIT). LDAP V3 is specified in RFC 2251 and is specifically targeted at management and browser applications that provide read/write interactive access to directories. HCD makes IODF data accessible via LDAP using the IBM Tivoli Directory Server for z/OS.

Local system name

When defining an XMP processor, you can specify an optional local CPC designator. If you do not specify a local system name, and a CPC name is given, the local system name defaults to the CPC name.

Logical control unit

A logical control unit (LCU) can be a single CU with or without attached devices or a group of one or more CUs that share devices. In a channel subsystem, a logical CU represents a set of CUs that physically or logically attach I/O devices in common. A logical CU is built from the information specified in the CU definitions. The physical CUs the device is attached to form part of a logical CU.

Logically partitioned (LPAR) mode

A central processor complex (CPC) power-on reset mode that enables use of the PR/SM feature and allows an operator to allocate CPC hardware resources (including central processors, central storage, expanded storage, and channel paths) among logical partitions. Contrast with *basic mode*.

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.

Master IODF

A master IODF is a centrally kept IODF containing I/O definitions for several systems or even for a complete enterprise structure. Master IODFs help to maintain consistent I/O data within a system and can provide comprehensive reports. From the master IODF subset IODF may be

generated to serve as production IODFs for particular systems within the structure.

MCF See *master configuration file*.

Migration

Refers to activities that relate to the installation of a new version or release of a program to replace an earlier level. Completion of these activities ensures that the applications and resources on your system will function correctly at the new level.

Multiple Image Facility (MIF)

A facility that allows channels to be shared among PR/SM logical partitions in an ESCON or FICON environment.

Multiple exposure device

A multiple exposure device is allocated by a single device number, but accessed by several device numbers, whereby 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-user access

Users can define the multi-user access attribute for IODFs so that multiple users can simultaneously update this IODF. An IODF is kept in exclusive update mode only for the duration of a single transaction. If the updates of the transaction are committed, another user may update the IODF without requiring the first user to release it. Though a user's changes are not immediately refreshed in the views of the other users, each user has a consistent view of the data either from the initial access to the IODF or after each last update that he had applied to the IODF.

MVS system

An MVS image together with its associated hardware, which collectively are often referred to simply as a system, or MVS system.

MVSCP

MVSCP (MVS configuration program) is the program that defines the I/O configuration to MVS. For this definition, information about devices, **EDTs**, and **NIP** consoles is required.

NIP console

A NIP (nucleus initialization program) console is a device that NIP uses as a console to display system messages. To define a device as a NIP console, it must first be defined to the channel subsystem and the current operating system (OS) configuration.

PCHID

See *physical channel identifier*.

PCIe See *Peripheral Component Interconnect Express*.

Peer coupling channel

A peer coupling channel is a coupling channel operating in peer mode, which means it can be used as a sender and receiver at the same time. It may be shared by several logical OS partitions (such as CF sender channels) and by a CF logical partition. In addition, peer channels provide more buffer sets and channel bandwidth than their counterparts. Peer channels are supported only on zSeries 900 servers and their successors.

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*.

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.

PPRC See peer-to-peer remote copy.

Preference value

Preference value is the value that is assigned to each **generic**. This value determines the sequence of allocation. The generics and the associated values are system-defined. The predefined order can be changed by means of the preference value.

Processor Resource/Systems Manager (PR/SM)

The feature that allows the processor to use several OS images simultaneously and provides logical partitioning capability. See also *LPAR*.

Production IODF

The production IODF is used by MVS/IPL to build **UCBs** and **EDTs**. It is also used to build **IOCDs** and IOCP input data sets. Several users can view a production IODF concurrently and make reports of it, but it cannot be modified. The production IODF that is used for IPL must be specified by a **LOADxx** member. The **LOADxx** member can reside either in **SYS1.PARMLIB** or **SYSn.IPLPARM**. If the **LOADxx** member resides in **SYSn.IPLPARM**, then **SYSn.IPLPARM** must reside on the IODF volume. If the **LOADxx** member resides in **SYS1.PARMLIB**, then **SYS1.PARMLIB** can reside on either the system residence (**sysres**) volume or the IODF volume.

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.

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.

SNA address

The system network architecture (SNA) address is a means to identify the support element of a CPC configured in a z Systems cluster. It consists of the network name (the network identifier of the LAN the support element of a CPC is connected to), and the system name (the identifier of the CPC within the network). In HCD, used as part of a processor definition for a CPC, the SNA address provides the association of a processor defined in the IODF with a CPC configured in a z Systems cluster.

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.

STP link

See Server Time Protocol link.

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.

Sysplex

A set of operating systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. See also *MVS system*.

z Systems cluster

A z Systems cluster is a configuration that consists of CPCs (central processor complexes), one or more Hardware Management Consoles, and may have one or more coupling facilities. The support elements that are attached to the CPCs are connected to a network. A Hardware Management Console connected to the same network allows the system operator to configure the CPCs, observe and control hardware operations, and perform software functions.

UCB Unit control block

UIM UIMs (unit information modules) perform the device-dependent part of the operating system configuration definition. There is a UIM for each supported device or device group. Each UIM recognizes and processes the values coded for its device or device group. HCD routines load all UIMs, either IBM or customer supplied, into virtual storage and make calls to the UIMs:

- During initialization
- During processing of an Add device or Change device request
- During generation of a print report
- During IPL

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.

VIO VIO (virtual input/output) is the allocation of data sets that exist in paging storage only. Only DASDs are eligible for VIO. Data sets are allocated to a paging device instead of to a real device.

Virtual channel identifier (VCHID)

For internal channel paths there does not exist a physical correspondence to hardware, hence 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 IODF

The work IODF is used to update an I/O definition and reflects the most recent status of the hardware configuration. After you have completed the updates, you can use the work IODF to create a production IODF. While you can update a work IODF and generate reports from it, it cannot be used to build UCBs and

EDTs, nor can it be used to generate an IOCDs, or an IOCP input data set.

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

Special characters

- *\$HCD\$ 222
- *\$HCDC\$ 222
- >peer-to-peer remote copy 136
- >PPRC 136

Numerics

- 4-character model 221
- 8-character control unit 221

A

- abend code
 - 00F 470
 - other than 00F 471
- abnormal termination 470
- access authority 359
- access list 110
- access to HCD services 507
- accessibility 531
 - contact IBM 531
 - features 531
- action bar 55, 393
- action code 60, 393
- action list 59
- action-to-action processing 59
- actions on objects, migration
 - enhancements 314
- activate
 - IOCDS, batch 338
- activate messages 236
- activating
 - configuration dynamically 230
 - switch configuration 183
 - sysplex-wide 234
 - trace 491
- activation actions available from CPC
 - Image List 242
- active IODF, view 230
- active switch 181
- active sysplex member list 235
- activity log 543
 - naming convention 32
- Activity Log panel 51
- activity logging 50
 - automatic 22, 51
 - browsing 270
 - HCD profile 22
 - printing 270
 - rules 52
 - volume serial number 22
- Add Device
 - peer-to-peer remote copy 136
 - PPRC 136
- adding
 - configuration package object 48
 - configuration packages 47
 - device to esoteric device list, migration 317

- adding (*continued*)
 - objects through migration 313
- advantages of extended migration 291
- agent 505
- all-character readable token 221
- ALLOC_SPACE 23
- allocating output data sets
 - volume serial number 22
- allow hardware deletes 232
- alter access 360
- alternate DD names 358
- analyze errors
 - batch migration 290
 - dialog migration 289
 - HCD problems 469
- API provided by HCD 499
- application programming interface 499
- assembler check, migration 289, 322
- assembler statements 282
- assembly listing 323
- assigning devices to esoterics 83
- assistive technologies 531
- associated data sets 32
- associated with processor and partition, migration 288
- attended export 41
- attribute group change 144
- attended mode 187
- AUTO_CHPID_EXCLUDE 189
- AUTO_CHPID_INCLUDE 190
- AUTO_MATCH_CU_DEVNUM 190
- AUTO_SS_ALTERNATE 190
- AUTO_SS_DEVNUM_SCHEME 190
- AUTO_SUG_CU_RANGE 191
- AUTO_SUG_DEV_RANGE 191
- AUTO_SUG_DYN_CHPIDIS 191
- AUTO_SUG_LPGROUP 192
- AUTO_SUG_OSGROUP 192
- AUTO_SUG_STAT_CHPIDIS 191
- AUTO_SWAD_EXCLUDE 192
- AUTO_SWAD_INCLUDE 192
- auto-assign 128
- autoconfiguration
 - overview 196
- autoconfiguration operation mode
 - attended mode 187
 - fast-path mode 187
 - unattended mode 187
- autoconfiguration policies
 - AUTO_CHPID_EXCLUDE 189
 - AUTO_CHPID_INCLUDE 190
 - AUTO_MATCH_CU_DEVNUM 190
 - AUTO_SS_ALTERNATE 190
 - AUTO_SS_DEVNUM_SCHEME 190
 - AUTO_SUG_CU_RANGE 191
 - AUTO_SUG_DEV_RANGE 191
 - AUTO_SUG_DYN_CHPIDIS 191
 - AUTO_SUG_LPGROUP 192
 - AUTO_SUG_OSGROUP 192
 - AUTO_SUG_STAT_CHPIDIS 191
 - AUTO_SWAD_EXCLUDE 192

- autoconfiguration policies (*continued*)
 - AUTO_SWAD_INCLUDE 192
 - changing between controller discoveries 192
- autoconfiguration policy 188
- autoconfigured control units 196
- autoconfigured I/O devices 196
- automatic activity logging
 - rules 52
- automatic I/O configuration 187

B

- back-level IODF 44
- backup an IODF 39
- base 543
- base IODF 187
- batch jobs
 - generated by HCD 72
- batch utility
 - activate a production IODF 338
 - build configuration data 339
 - build CONFIGxx member 339
 - build IOCDS or IOCP input data set 335
 - build JES3 initialization stream checker data 339
 - build production IODF 334
 - build work IODF 335
 - compare IODFs or CSS/OS reports 348
 - copy IODF 342
 - create graphical configuration reports 346
 - description 327
 - example for copy IODF 355
 - example for migration 290
 - EXEC procedures 30
 - export IODF 353
 - FCP device data 339
 - functions 327
 - import IODF 352
 - initialize IODF 329
 - input parameter string 328
 - job steps during transmit procedure 480
 - migrate an input data set 290, 331
 - print configuration reports 343
 - starting the dialog 328
 - upgrade IODF 330
- BCPIi
 - working with CPC images 246
- BCTC 157
- blocking ports 178
- BookMaster 257
- BOTTOM command 56
- browse activity log 270
- build
 - CONFIGxx member, dialog 247
 - CONFIGxx, batch 339
 - FCP device data, batch 339

- build (*continued*)
 - HCPRIO input data set, batch 338
 - IOCDS, batch 335
 - IOCDS, dialog 212
 - IOCP input data set 221
 - IOCP input data set, batch 335
 - IOCP input data set, dialog 219
 - JES3, batch 339
 - OS configuration data set 226
 - production IODF, batch 290, 334
 - production IODF, dialog 208
 - work IODF, batch 335
 - z Systems cluster IOCDSs, dialog 215
- build configuration statements 291

C

- candidate list 111
- cascade switching 163
- cascaded FICON fabric 130
- catalog
 - considerations 365
 - scan utility< 366
- cataloged data set 365
- CBDCHCD 17, 18
- CBDMGHOM routine 499
- CBDMLPLG
 - define HCD LDAP backend as a plug-in 376
- centralized navigation 74
- CF channel path, migration enhancements 306
- CF duplexing 10
- CFReport 224, 543
- chained switches 164, 174
- change log 543
 - naming convention 32
- change log data set
 - allocation space extension 23
- change log file 494
 - name 51
 - size 51
- CHANGE_LOG 22
- changing
 - channel path attributes, migration 317
 - channel paths 120
 - configuration packages 47
 - consoles 151
 - control unit and device attributes, migration 317
 - control unit attachment parameters for multiple processors 132
 - control units 130
 - coupling facility connection 123
 - CSS-related definitions for a group of devices 143
 - device number 147
 - devices 141
 - DYNAMIC parameter for a group of devices 144
 - EDT ID 81
 - EDTs 81
 - esoterics 83
 - esoterics for a group of devices 143
 - IODF attributes 41

- changing (*continued*)
 - LOCANY parameter for a group of devices 144
 - operating mode of channel paths 122
 - operating system configuration 79
 - OS configuration ID 79
 - OS-related definitions for a group of devices 143
 - partition name, migration 317
 - partitions 99
 - ports from installed to uninstalled 176
 - ports from occupied to not occupied 176
 - processor 87
 - processor ID 89
 - processor type 87
 - production IODF 38
 - switch configuration ID 179
 - switch configurations 169
 - switch data 169
 - switches 169
 - to another IODF 38
 - type of CF channel paths 121
 - type of channel paths 121
 - type/model of a group of devices 145
- changing channel path ID 121
- channel path
 - access list 110
 - aggregating 124
 - candidate list 111
 - changing 120
 - CHPID statement 305
 - compare report, example 459
 - connectivity report, example 407
 - defining 109
 - deleting 126
 - detail report, example 406
 - migration enhancement 305
 - operation mode 108
 - spanned 116
 - summary report, example 404
 - working with 108
- channel path ID
 - changing 121
- channel subsystem 93, 543
 - changing 96
 - copy 94
 - copying 96
 - defining 93
 - deleting 97
- channel subsystem summary report, example 400
- CHID 110
- CHID function
 - summary 402
- CHLOG_EXTENSION 23
- CHPID 543
 - spanned 116
- CHPID Mapping Tool 224, 543
- CHPID report 257
- CHPID statement 305
- CIB channel 115
 - over-defining 116
- client 505
 - HCD client 505

- CLIST used by HCD 18
- CNC channel 156
- CNC side information 159
- CNTLUNIT statement 308
- coexistence considerations 53
- color settings for graphical reports 25
- combined input data sets 288
- command
 - syntax diagrams xviii
- command line 64
- commands
 - FIND 179
 - GOTO 66
 - HELPIID 474
 - HELPTTEST 474
 - how to use 64
 - INSTRUCT 56
 - LEFT 56
 - LOCATE, graphical report 262
 - PANELID 473
 - PFSHOW 64
 - REFRESH, graphical report 263
 - RIGHT 56
 - SAVE 261, 263
 - SHOWIODF 41
 - SHOWMSG 487
 - SHOWMSG ACTIVATE 233
 - TRACE 491
 - TRACE, HCM 494
- compare
 - CONFIGxx members 249
 - CSS/OS views, dialog 269
 - IODFs or CSS/OS Reports, batch 348
 - IODFs, dialog 265
- compare reports
 - channel path 459
 - control unit 461
 - control unit attachment 460
 - CSS / OS device 468
 - device 462
 - device attachment 461
 - esoteric 465
 - Operating System Compare Report 466
 - OS console 467
 - OS device 467
 - partition 458
 - PCIe function 456
 - processor 455
 - switch 463
 - switch configuration detail 464
 - switch detail 463
- compatibility 53
- component ID 469
- concatenation 15
- configuration data
 - activating dynamically 230
 - consolidating 74
 - define 73
 - definition sequence 76
 - delete 73
 - modify 73
 - process 207
 - validation 3
- configuration package
 - add 48
 - define 47

- configuration package (*continued*)
 - delete 48, 49
 - edit 47
 - list 46, 48
 - merge 48
 - problem determination for transmit procedure 480
 - replace 48
 - transmit 49
 - work with 46, 48
- configuration report
 - build with batch 339
- configuration reports
 - CF channel path connectivity report, example 407
 - channel path detail report, example 406
 - channel path summary report, example 404
 - channel subsystem reports, example 399
 - control unit detail report, example 409
 - control unit summary report, example 409
 - CSS report, description 253
 - CTC connection 254
 - CTC report, example 422
 - device detail report, example 412
 - device summary report, example 412
 - eligible device table report, example 419
 - examples 399
 - graphical configuration report 257
 - I/O definition reference 255
 - I/O definition reference, example 451
 - I/O path report 423
 - I/O path, description 254
 - IOCDS report, example 403
 - LCU report, example 454
 - MVS device detail report, example 418
 - MVS device report, example 416
 - NIP console report, example 420
 - operating system report, description 254
 - operating system report, example 416
 - operating system summary report, example 416
 - partition report, example 403
 - PCIe Function Summary Report 400
 - print with batch 343
 - printing graphical reports 257
 - printing textual reports 255
 - problem determination 475
 - processor summary report, example 400
 - supported hardware report 425
 - supported hardware, description 255
 - Switch Configuration Compare Report 464
 - switch configuration detail report 416
 - switch configuration summary report, example 415
- configuration reports (*continued*)
 - switch detail report, example 415
 - switch report, description 254
 - switch report, example 414
 - switch summary report 415
 - viewing graphical reports 257
 - VM console report, example 421
 - VM device detail report, example 421
 - VM device report, example 420
- configuration status 237
- CONFIGxx member
 - build using dialog 247
 - build with batch 339
 - compare using dialog 249
- connecting
 - another switch to a switch 174
 - channel path to switch 172
 - control unit to switch 173
 - coupling facility channel paths 117
- connection table 245
 - DD name HCD_CNTL 246
 - reachable system 240
 - record format 246
- connections of switches, possibilities 163
- console
 - changing 151
 - configuration report, description 254
 - configuration report, example 420
 - deleting 151
 - report, example 421
 - working with 150
- consolidating configuration data 74
- contact
 - z/OS 531
- context menu 59
- control unit
 - attachment compare report, example 460
 - attachment data to processor 128
 - changing 130
 - changing attachment parameters for multiple processors 132
 - CNTLUNIT statement 308
 - compare report, example 461
 - configuration report, example 409
 - defining 126
 - deleting 133
 - disconnecting 132
 - migration enhancement 308
 - priming, serial number 133
 - supported by system 274
 - working with 126
- control unit list 75
- control unit model 280
- control units
 - autoconfigured 196
- Converged Ethernet (RoCE) 103
- copying
 - IODF, batch example 355
 - IODF, batch utility function 342
 - IODF, with dialog 40
- coupling facility 543
 - changing connection 123
 - connecting channel paths 117
 - disconnecting channel paths 120
 - overview 9
- coupling facility (*continued*)
 - view information 155
- coupling facility connection report 257
- CPC (central processor complex)
 - support in HCD 11
- CPC (central processor complex) 543
 - specify name 86
- CPC image
 - connection table 245
 - reachable system 240
- CPC Image List
 - available activation actions 242
- CPC images
 - BCPii 246
 - dynamic activation of I/O changes 239
 - operation status 239
 - prerequisites 246
 - RACF Facility Class profiles 360
 - support element (SE) 246
- Create Work I/O Definition File panel 33
- create work IODF from production IODF 38
- creating
 - graphical configuration report, batch 346
 - graphical configuration report, dialog 257
 - IODF using batch 290
 - IODF using dialog 32
 - JES3 initialization stream checker data 226
 - new objects 73
- cross operating system support 5
- CSS
 - changing 96
 - copy 94
 - copying 96
 - defining 93
 - deleting 97
- CSS (channel subsystem)
 - configuration report, description 253
 - configuration report, example 399
 - group change action 143
 - overview 1
- CSS-related definitions for devices 137
- CSS/operating system device compare report, example 468
- CSS/OS report, compare with batch 348
- CTC channel 156
- CTC connection 157
- CTC Connection List 157
- CTC connection report
 - filter 160
 - incomplete definitions 159
 - print 160
 - report, example 422
 - with MIF 157
 - with shared channels 157
- CTC Connection Report 254
- CTC connection report>Select one or more CTC connections and press the Enter key to display diagnostic messages for the connections.
 - diagnostic messages 160

- CTC connections
 - copy/repeat 95
 - update 95
- CTC side information 159
- CU report 257
- CUADD, CTC connection 157
- customizing HCD 15

D

- D/R OS configuration 78
- D/R site OS configurations 78
- data entry dialog 56
- data sets
 - allocation for migration 325
 - cataloged with esoteric name 365
 - combined 288
 - example 222
 - generated during transmit configuration package 481
 - migrate with batch 331
 - migration 279
 - migration using the dialog 287
 - relationship in HCD 7
- data sets due to HCM requests
 - allocation space 23
- data space 26
- DCF 257
- DCM 113
- ddnames
 - alternate 358
 - default for migration 325
 - standard 357
- dedicated connection 163, 178
- dedicated operation mode 108
- default connection 177
- default switch matrix 179
- defining
 - configuration packages 47
 - connections to switches 172
 - switch configurations 177
 - switches 166
- device
 - activate messages 237
 - attachment compare report, example 461
 - changing 141
 - changing CSS-related definitions for a group 143
 - changing DYNAMIC parameter 144
 - changing esoterics for a group 143
 - changing LOCANY parameter 144
 - changing number 147
 - changing OS-related definitions for a group 143
 - changing type/model of a group of devices 145
 - channel paths 126
 - compare report, example 462
 - compare, CSS/operating system 468
 - configuration package object 49
 - configuration packages 48
 - configuration report, description 254
 - console list, migration 317
 - consoles 151
 - control unit, migration 316
 - control units 133

- device (*continued*)
 - defining 134
 - defining a group in one step 137
 - defining CSS-related definitions 137
 - defining OS-related definitions 140
 - defining SS-related definitions 137
 - deleting 150
 - detail report, example 412
 - device from esoteric device list, migration 317
 - device, migration 316
 - devices 150
 - disconnecting from operating system 147
 - EDTs 82
 - esoteric device list, migration 317
 - esoterics 85
 - identification 137
 - IODEVICE statement 310
 - IODF 40
 - migration enhancement 310
 - operating system configuration 80
 - partition, migration 316
 - partitions 102
 - priming, serial number 149
 - priming, volume serial number 149
 - processors 92
 - protection 364
 - restricting partition access 139
 - summary report, example 412
 - supported by system 275
 - switch configuration 171
 - switch-to-switch connection, migration 317
 - switches 171
 - switches, confirm 171
 - working with 134
- device candidate list
 - empty 139
 - null 139
- device list 75
- device lists
 - locate processing 262
- device serial number 137
- device type group change 145
- diagnostic information and tools 487
- diagnostic message, CTC connection 160
- diagnostic messages 422
- dialog, usage 55
- different IODF levels 53
- disable write protection 215
- disabled marker 176
- disconnecting
 - control units from a processor 132
 - coupling facility channel path connections 120
 - devices from operating system 147
 - multiple control units from a processor 132
- discovery
 - scope 196
- dispatcher 505
- display active sysplex member list 235
- display diagnostic message, CTC connection 160
- display graphical reports 260
- dualwrite option 212

- dumps 494, 496
- duplicate control unit 283
- duplicate device 285
- dynamic channel path management 113
- dynamic connection 156, 163
- dynamic connection ports 178
- DYNAMIC parameter 144
- dynamic partition 99
- dynamic reconfiguration management
 - activation 230
 - failure, sysplex-wide 237
 - glossary definition 544
 - hardware and software changes
 - allowed 231
 - identify device 137
 - overview 4
 - profile access 359
 - rejection 233
 - resource access 359
 - software only changes allowed 234
 - sysplex-wide 234
- dynamic switch 163

E

- editable fields 61
- EDT 544
 - changing 81
 - configuration report, description 254
 - configuration report, example 419
 - defining 80
 - deleting 82
 - EDT statement 295
 - migration enhancement 295
 - repeating 82
 - working with 80
- EDT statement 295
- EMIF
 - see MIF 544
- enable write protection 215
- ending HCD 18
- enhanced migration 291
- entry port, CTC connection 156
- entry switch 163
- environment for HCD 8
- error message, CTC connection 159, 160
- errors
 - diagnosis 469
 - during IPL 472
 - during migration 289, 324
 - during transmit procedure 480
- ESCON 544
- ESCON director, CTC connection 157
- ESCON Manager 9, 544
- ESCON Manager lock 183
- esoteric compare report, example 465
- esoterics 544
 - changing 83
 - defining 83
 - deleting 85
 - device group name 365
 - keyword to allow mixed 23
 - migration enhancement 296
 - repeating 84
 - token 296, 365
 - token, migration 282
 - UNITNAME statement 296

- esoterics (*continued*)
 - working with 82
- esoterics token 366
- establishing coupling facility channel
 - path connections 117
- establishing dedicated connections 178
- EXEC procedures 30
- exit, function key 65
- explicit device candidate list 101, 139
- export
 - IODF, batch 353
 - IODF, dialog 41
- EXPORTED_HLQ 23
- extended IQD 114
- extended migration 291
 - IOCP input data set 221

F

- fabric 163
- Facility Class profiles
 - for working with CPC images 360
- failure, diagnosis 469
- fast path 66
- fast-path mode 187
- FCP device data 227
- FCP SAN configuration template
 - file 227
- FCTC 156
- FICON 544
 - cascade switching 163
 - switch fabric 163
- FICON Channel-to-Channel support 156
- FICON devices
 - I/O Autoconfiguration 187
- FICON switch
 - downgrading link address 130
 - upgrading link address 130
- FICON switches 163
- fields
 - editable 61
 - input and output 57
 - promptable 63
- filter CTC connection 160
- filter mode 71
- filtering a list 69
- FIND command 179
- font for graphical reports 25
- formatting type for graphical reports 25
- function keys
 - actions 55
 - add 73
 - backward 56
 - command 64
 - Enter 65
 - exit 65
 - forward 56
 - help 64
 - how to use 64
 - instruct 56
 - jump 261
 - left 56
 - previous 65
 - problem determination 473
 - prompt 57
 - reset 61
 - right 56

- function keys (*continued*)
 - setting 24 keys on 18
 - split 367
 - zoom 261
- FUNCTION statement 304

G

- GDDM
 - installation 15
 - usage for reports 257
- generate switch matrix 179
- generic device type 544
- generics 82
 - migration enhancement 297
 - UNITNAME statement 297
- GML 257
- GOTO command 66
- graphical configuration report
 - create with batch 346
 - example 454
 - font 25
 - jumping to action lists 261
 - LCU report, example 454
 - limiting a graphical report 259
 - prerequisites 257
 - printing 257
 - problem determination 475
 - scaling factor 24
 - viewing 257
- graphical navigation 74
- graphical representation 257
- group change action for control
 - units 129
- group changes
 - attribute group change for
 - devices 144
 - CSS group change for devices 143
 - device type group change 145
 - esoterics for a group of devices 143
 - OS group change for devices 143

H

- hardware and software changes
 - allowed 231
- Hardware Management Console 11, 544
- hardware report 425
- hardware support 271
- HCA 115
- HCD (Hardware Configuration Definition)
 - abnormal termination 470
 - and coupling facility 9
 - and I/O Operations (ESCON Manager) 9
 - customizing 15
 - data sets relationship 7
 - differences between HCD and
 - MVSCP/IOCP 1
 - ending 18
 - environment 8
 - installation considerations 15
 - object management services 499
 - overview 1
 - overview of functions 4

- HCD (Hardware Configuration Definition) (*continued*)
 - product identifier 469
 - setting up 15
 - starting 18
- HCD agent 505
- HCD client 505
- HCD dispatcher 505
- HCD LDAP backend 369
- HCD LDAP backend configuration 377
- HCD LDAP schema 377
- HCD profile 19, 222
 - standard keywords 19
- HCD Profile Options dialog 20
- HCD profile, define 509
- HCD services
 - access 507
- HCDCNTL
 - DD name for connection table 246
- HCM change log data set
 - allocation space extension 23
- HCM master configuration file 545
- HCM MCF
 - volume serial number 22
- HCPRIO input data set 544
 - build with batch 338
 - build with dialog 226
 - migration 285
- help
 - how to get 64
 - problem determination 474
 - testing panels 474
- HELPID command 474
- HELPTTEST command 474
- hiding parameter/feature
 - definitions 148
- hierarchical navigation 74
- highlighting 57
- HLQ
 - for exporting IODFs 23
- HMC 11
- HMC-wide activation 29
- HOM services 499
 - example 503
 - functions 502
 - input and output areas 500
 - invocation 499
 - request block (HRB) 501
 - return codes 504
- host access
 - IPv6 505
 - TCP/IP protocol 505
- host communication adapter 115
- HSA 235
- HSA token 215
- HWI
 - RACF Facility Class profiles 360
- HWI profiles
 - access rights 362

I

- I/O Autoconfiguration 187
 - base IODF 187
 - enhancements xxv
 - fast-path 203
 - invocation 196

- I/O Autoconfiguration (*continued*)
 - operational considerations 203
 - policy keywords 189
 - scope of discovery 196
 - target IODF 187, 198
 - unattended mode 203
 - updates to the proposals 199
 - I/O cluster 545
 - I/O concurrency level 101, 129, 131, 411
 - I/O configuration definition reference
 - configuration report, example 451
 - description 255
 - print 343
 - I/O device list
 - PPRC usage 135
 - I/O devices
 - autoconfigured 196
 - I/O Operations 545
 - I/O Operations, overview 9
 - I/O path list 229
 - I/O path report
 - description 254
 - example 423
 - IBM ES/4381
 - build IOCP input data set 219
 - IBM Tivoli Directory Server for z/OS 369
 - address space 485
 - ID statement 300
 - identifying problems 469
 - IEC161I (copy IODF batch utility) 356
 - IEFES0JL (catalog scan utility) 366
 - image number, CTC connection 157
 - import
 - IODF, batch 352
 - IODF, dialog 44
 - incomplete CTC definitions 159
 - increased system availability 4
 - incremental update 313, 314
 - initialization problems 477
 - initialization stream checker data for JES3 226
 - initialize IODF from VSAM DIV file 329
 - input fields 57
 - input parameter string, batch utilities 328
 - input to stand-alone IOCP 220
 - Input/Output Subsystem 187
 - installation-static 281
 - installed port range 168
 - installed ports 176
 - installed UIMs 276
 - installing
 - GDDM 15
 - HCD 15
 - INSTRUCT command 56
 - instruction area 56
 - insufficient data set sizes 325
 - interface
 - batch utility functions 327
 - dialog 55
 - HCD object management services 499
 - Internal Shared Memory PCIe Adapter (ISM). 103
 - Internet Protocol
 - Version 6 505
 - introduction 1
 - invoking batch utility functions 327
 - IOCDS 544
 - build with batch 335
 - build with dialog 212
 - build z Systems cluster IOCDSs 215
 - report, example 403
 - token 215
 - use with HCD 5
 - write in preparation for processor upgrade 212
 - IOCONFIG statement 294
 - IOCP 545
 - build input data set with batch 335
 - build input data set with dialog 219
 - example 222
 - extended migration 221
 - migrating additional data sets 283
 - migrating input data sets 279
 - stand-alone 220
 - use with HCD 1
 - IODEVICE statement 310
 - IODF 545
 - activity log 52
 - available IODFs 58
 - change log 52
 - multi-user access status 58
 - V5 IODF 40, 45
 - validated work IODF 225
 - IODF (I/O definition file)
 - activity logging 33, 50
 - attributes 32
 - backup 39
 - build production IODF, batch 334
 - build production IODF, dialog 208
 - build work IODF, batch 335
 - change attributes 41
 - change during session 38
 - compare 265
 - compare reports, example 454
 - compare with batch 348
 - copy with batch 342
 - copy with dialog 40
 - data model 511
 - data space 26
 - delete 40
 - description 2, 31
 - different levels 53
 - dump 494
 - export with batch 353
 - export with dialog 41
 - import with batch 352
 - import with dialog 44
 - initialize 329
 - LDAP support 369
 - maintenance 39
 - master 36
 - multiple 35
 - naming conventions 31
 - object classes and attributes 512
 - objects managed in 6
 - prefix 251
 - release level compatibility 53
 - repair 495
 - sharing 34
 - single 35
 - space allocation 33
 - IODF (I/O definition file) (*continued*)
 - specify 32
 - subset 36
 - suffix 251
 - transfer to a different system or location 496
 - upgrade with batch 330
 - upgrade with dialog 44
 - used at IPL 6
 - used for last IPL 38
 - view currently accessed one 41
 - volume serial number 33
 - IODF checker 27, 209
 - IOS 187
 - IPL 545
 - attribute list 218
 - attribute management functions 11
 - errors 472
 - IODF processing 251
 - manage attributes for z Systems cluster 218
 - specify an IODF 251
 - with MVSCP 4
 - IPLADDR value 218
 - IPLPARM value 218
 - IPv6 505
 - IQD channel
 - IEDN access 114
 - IQDX function 114
 - IQDC frame size 114
 - IQDX 114
 - ISPF considerations 367
 - ISPF list file 471
 - ISPF table 181
 - ISPF, tailoring 18
- ## J
- JCL
 - for creating a user ID for UNIX System Services 509
 - JES3 initialization stream checker data 226
 - JES3, build with batch 339
 - job statement information 72
 - jumping from graphical reports 75, 261
- ## K
- keyboard
 - navigation 531
 - PF keys 531
 - shortcut keys 531
 - keywords
 - I/O Autoconfiguration 189
 - keywords in HCD profile 22
- ## L
- LCSS 85, 108, 545
 - LCU (logical control unit) 545
 - report, example 454
 - report, how to generate 257
 - LDAP
 - CBDMLPLG plug-in 376

- LDAP (*continued*)
 - IBM Tivoli Directory Server for z/OS
 - address space 485
- LDAP (lightweight directory access protocol) 545
 - address space, HCD instance 485
 - error in HCD LDAP backend 486
 - HCD instance startup fails 486
 - HCD LDAP backend debug
 - level 380
 - HCD LDAP backend,
 - customizing 377
 - interface capability 5
 - operational behavior 382
 - plug-in abnormal termination 485
 - problem determination 484, 485
 - return code 486
 - server, ds.conf file parameters 377
 - support 369
 - transactions 387
- LDAP response messages 488
- LDAP schema 377
- LEFT command 56
- legend, CTC report 422
- library concatenation 15
- limit CTC connection report 256
- limiting a graphical report 259
- link address 129
- link address, CTC connection 156
- lists
 - action list 59
 - configuration package list 46
 - configuration package object list 48
 - filtering 69
 - I/O path list 229
 - numbered selection lists 57
 - selection methods 57
 - unnumbered multiple selection
 - lists 58
 - unnumbered single selection lists 57
 - working with 57
- LOAD parameter 251
- local system name 86, 545
- LOCANY parameter 144
- LOCATE command 56
- LOCATE command, graphical
 - report 262
- locate processing on device lists 262
- logging 50
- logical address 129
- logical partition group 193
- LP group 192, 193
- LPAR (logical partitions) 545
 - defining 97

M

- maintain IODF 39
- manage z Systems cluster IPL
 - attributes 218
- MAP_CUTYPE 26, 280
- mapping of migrated data 282
- master configuration file 545
 - volume serial number 22
- master IODF 36, 545
 - merge changes 48
- matrix, generate switch 179

- maximum frame size 114
- MCF 545
 - volume serial number 22
- MCF data set
 - allocation space extension 23
 - naming convention 32
- MCF_EXTENSION 23
- message list 62, 322
- Message List
 - delete messages 63
 - explain messages 63
 - save messages 63
- message list, CTC connection 160
- message log 487
- messages
 - overview 487
 - reason code 470
- MIF 546
 - defining with HCD 108
- MIF, CTC connection 157
- Migrate IOCP / MVSCP / HCPRIO Data
 - dialog 287
- migrate switch configuration 181
- MIGRATE_EXTENDED 222
- migration
 - additional input data sets 282
 - advantages 291
 - assembler check 289
 - batch utility 331
 - changing input data sets 280
 - conflicts 280
 - consideration 280
 - control unit type mapping 26, 280
 - coupling facility updates 289
 - data set allocation 325
 - enhancements 291
 - errors 289
 - existing input data sets 279
 - existing switch configurations 181
 - extended migration 291
 - for combined IOCP/MVSCP input
 - data set 279
 - log 324
 - LPAR considerations 279
 - multiple IOCP/MVSCP
 - statement 279, 282
 - new parameters 291
 - overview 13
 - partial 313
 - preparation 280
 - prerequisites for extended
 - migration 222
 - processing mode 288
 - protocol support for control
 - units 281
 - return code 289
 - roadmap 13
 - sequence 279
 - solving errors 322
 - termination 289
 - using the HCD batch utility 290
 - using the HCD dialog 287
 - validation rules 280
- migration log 487
- minimum installed port range 298
- mixed environment 226

- mixed esoterics
 - profile keyword 23
- module prefix 469
- moving directly 66
- multi-user access 33, 151
 - abend 34
 - enforcing a refresh view on the
 - IODF 151
 - release a lock 34
 - scenarios 151
- multiple exposure device 546
- multiple IODFs 35
- multiple switch activation 184
- MVS device report, example 416
- MVS devices 426
- MVSCP (multiple virtual system
 - configuration program) 546
 - differences to HCD 1
 - migrating additional input data
 - sets 285
 - migrating input data sets 279
 - use with HCD 1

N

- naming convention of IODF 31
- navigating through HCD 74
- navigation
 - keyboard 531
- navigation map 75
- network name 86
- next POR 249
- NIP console 546
 - NIPCON statement 295
 - working with 150
- NIP console report, example 420
- NIPCON statement 295
- Notices 535
- null device candidate list 139
- numbered selection lists 57

O

- occupied indicator
 - PORT statement 298
- operating mode
 - changing 122
 - how to use 108
- operating requirements for HCD 8
- Operating System Compare Report,
 - example 466
- operating system configuration
 - changing 79
 - defining 77
 - deleting 80
 - group change action 143
 - IOCONFIG statement 294
 - migration enhancement 294
 - repeating 79
 - report, description 254
 - reports, example 416
 - working with 77
- operation mode
 - attended mode 187
 - unattended fast-path mode 187
- operation status of CPC images 239

- optional comment for SAVE command 264
- options for text reports 24
- OS configuration
 - D/R site OS configuration 78
- OS configuration group 194
- OS console compare report, example 467
- OS device compare report, example 467
- OS group 192, 194
- OS group change action 143
- OS parameters/features 312
- OS report 254
- OSD channel 115
- output field 57
- overview of HCD 1
- overview, migration 13

P

- panel
 - flow of HCD options 393
 - identifier 55, 473
 - layout 55
 - problem determination 473
 - title 55
- PANELID command 473
- partial migration 313
- partition
 - changing 99
 - compare report, example 458
 - configuration report, example 403
 - defining 97
 - deleting 102
 - dynamic I/O 99
 - migration enhancement 301
 - repeating 100
 - RESOURCE statement 301
 - transferring 100
 - working with 97
- partition access to CHPIDs 232
- partition usage 98
- PassTicket
 - for working with CPC images on z/VM 364
- PassTickets
 - for working with CPC images on z/OS 362
- PAV alias device 138
- PCHID 224, 547
- PCIe xxvi
- PCIe function
 - FUNCTION statement 304
 - migration enhancement 304
 - PCIe Function Compare Report 456
 - PCIe Function Summary Report 400
 - User Defined Identifier (UDI) 303
- PCIe Function Compare Report 456
- PCIe Function Summary Report 400
- PCIe functions 103
- peer coupling channel 546
- peer-to-peer communication 156
- peer-to-peer remote copy 547
- Peripheral Component Interconnect Express xxvi
- Peripheral Component Interconnect Express (PCIe) 103
- PFSHOW command 64
- physical channel identifier 224, 547
- physical network ID 114
- planning
 - migration consideration 280
 - migration, data set allocation 325
 - your configuration 73
- Point-to-point CTC connection 157
- policy keywords
 - autoconfiguration 189
- POR (power-on reset) 4
- POR, switch IOCDS 239, 250
- port configuration 299
- port list 175
- port matrix 163, 177
- port range 168
- PORT statement 298
- port, moving switch 170
- PORTCF statement 299
- ports, blocking 178
- PPRC 547
- PPRC usage
 - in I/O Device List 135
- PR/SM 10
- pre-migration activities 280
- preallocation 325
- preference value 82, 547
- preparing for migration 280
- prerequisites 9
 - working with CPC images 246
- previous, function key 65
- priming
 - control unit serial number 133
 - device data 149
 - processor serial number 92
 - switch port connection 174
 - switch port name 174
 - switch serial number 169
- printing
 - activity log 270
 - compare reports 265
 - configuration reports, batch 343
 - CTC connection 160
 - graphical configuration report 258
 - graphical reports 259
 - list panels 263
 - problem determination 475
 - textual configuration reports 253
- problem determination
 - abnormal termination 470
 - diagnostic information and tools 487
 - errors during IPL 472
 - graphical reports 257, 475
 - help information 474
 - identifying problems 469
 - initialization of HCD 477
 - IODF dump 494
 - panels and function keys 473
 - repair an IODF 495
 - system abend code '00F' 470
 - system abend code other than '00F' 471
 - textual reports 475
 - trace facility 488
 - transmit procedure 480
 - UIMS 478
- problem reporting data bases 496

- process configuration data 207
- processing mode, migration 288
- processor
 - changing 87
 - changing type 87
 - compare report, example 455
 - configuration report, example 400
 - defining 85
 - deleting 92
 - ID statement 300
 - migration enhancement 300
 - priming, serial number 92
 - repeating 89
 - support level 86
 - supported by system 271
 - working with 85
- Processor Resource/Systems Manager glossary definition 547
- processor upgrade, write IOCDS 212
- product identifier 469
- production IODF 547
 - build with batch 334
 - build with dialog 208
 - change 38
 - format of 32
 - how to specify 32
 - replace during build production IODF 208
- profile 222
 - HCD 509
- profile for HCD 19
- profile keywords
 - standard 19
- profile option
 - mixed esoteric 23
- profile options 20
- programming interface for HCD 327
- promptable fields 63
- protected devices 364
- protocol support for control units, migration 281

Q

- query supported hardware 271

R

- RACF 359, 361, 364
- RACF backend SDBM 369
- RDMA 103
- re-migrate IOCP input data sets 222
- reachable system 240
- read access 360
- reason code 470
- receiving an IODF 44
- reconfigurable operation mode 108
- REFRESH command, graphical report 263
- refreshing a view on IODFs 151
- refreshing the active sysplex member list 235
- Regional Crypto Enablement (RCE). 103
- rejection of dynamic activation 233
- relationship of data sets 7
- release level compatibility 53

- Remote Direct Memory Access (RDMA) 103
- repair an IODF 495
- repeating
 - action 74
 - channel subsystem 94
 - description 74
 - EDTs 82
 - esoterics 84
 - operating system configuration 79
 - partitions 100
 - processors 89
- replacing
 - during build production IODF 208
- replacing objects through migration 313
- reporting problems 496
- reports, text, options 24
- requirements
 - for setting up HCD 15
 - for using HCD 8
 - migrating IOCP/MVSCP data 280
- reserved partition 99
- reset function key 61
- reset source configuration
 - sysplex-wide 237
- RESOURCE statement 301
- restricting partition access for
 - devices 139
- resume activation sysplex-wide 237
- return code, migration 289
- RIGHT command 56
- roadmap, migration 13
- RoCE 103

S

- save
 - switch configuration 184
- SAVE command 261, 263
- SAVE command, CTC connection 160
- saved switch file 181
- scaling factor for graphical reports 24
- scenario
 - multi-user access 151
- schema file for LDAP 377
- schema.hcd.ldif 377
- scrolling 56
- SCTC 157
- SDBM
 - RACF backend 369
- SE
 - working with CPC images 246
- search order at IPL 251
- searching data bases 496
- security considerations 359, 364
- selection markers 59
- selection methods in lists 57
- sending an IODF 41
- sending comments to IBM xxi
- sensing 11, 423
- sequence of migration 279
- sequence to define a configuration 76
- serial number
 - control unit 133
 - device 149
 - processor 92
 - switch 169

- serial number for device 137
- server started task log 488
- Server Time Protocol 547
- Server Time Protocol (STP) link 120
- setting up HCD 15
- severity of messages 160
- shared channels, CTC connection 157
- shared operation mode 108
- sharing IODFs 35, 53, 365
- shortcut keys 531
- showing parameter/feature
 - definitions 148
- SHOWIODF command 41
- SHOWMSG ACTIVATE command 233
- SHOWMSG command 487
- single IODF 35
- single point of control 4, 234
- single point of failure xxviii
- single selection lists 57
- single switch activation 183
- SMP processor 547
- SMP processors
 - term definition 85
- SMS considerations 366
- SNA address 87
- software only changes allowed 234
- solving migration errors 322
- space allocation 33
- spanned channel paths 108
- spanned operation mode 108
- specifying
 - input data sets for migration 287
 - IODF, dialog 32
- split screen 367
- SPOF xxviii, 230
- SS (subchannel set) 138
- SS-related definitions for devices 138
- stand-alone CTC adapter 157
- stand-alone IOCP 220
- standard DD names 357
- standard profile keywords 19
- starting
 - HCD 18
 - migration 287
 - the dialog, batch 328
- status of configuration 237
- STP 547
- STP link 120
- subchannel set 93, 138
 - glossary definition 548
- summary of changes xxiii, xxiv
- summary reports
 - CHID function 402
 - PCIe function 400
- support element
 - working with CPC images 246
- support level 86
 - online description 87
 - provided functions 87
- supported hardware 271
- supported hardware report 425
- supported migration paths 14
- surrogate user ID
 - for exporting IODFs 43
- SWCONF statement 299
- switch
 - blocking ports 178

- switch (*continued*)
 - chained switches 164
 - changing data 169
 - compare report, example 463
 - confirming delete 171
 - connections to switches 172
 - dedicated connection 163
 - default connection 177
 - defining 166
 - defining connections 172
 - deleting 171
 - disabled marker 176
 - dynamic connection 163
 - dynamic switch 163
 - entry switch 163
 - graphical report 257
 - matrix, default 179
 - migrating 181
 - migration enhancement 297
 - moving ports 170
 - port matrix 163
 - port range 168
 - possible connections 163
 - priming, prerequisites 9
 - priming, serial number 169
 - report, example 414
 - supported by system 273
 - SWITCH statement 297
 - textual report, description 254
 - validation 165
 - working with 163
- switch configuration
 - activating 183
 - changing 169
 - changing ID 179
 - compare report, example 464
 - defining 177
 - definition 163
 - deleting 181
 - generate matrix 179
 - migrating 181
 - migration enhancement 299
 - report, example 415
 - save 184
 - select other 176
 - SWCONF statement 299
- Switch Configuration Compare Report 464
- switch IOCDS for next POR 212, 215, 249
- switch IOCDS for next POR,
 - sysplex-wide 239, 250
- SWITCH keyword 163
- switch port
 - priming, name and connection 174
- SWITCH statement 297
- symptom table 469
- syntax diagrams
 - how to read xviii
- SYS1.LOGREC record 471
- SYS1.SAMPLIB 366
- sysplex 234, 548
- sysplex couple data set 235
- sysplex member list 235
- sysplex, support 11
- system abend code
 - '00F' 470

system abend code (*continued*)
 other than '00F' 471
system names 235

T

target IODF 187, 198
TCP/IP definitions
 IPv6 505
termination
 of initialization 477
 of migration 289
test an activation sysplex-wide 236
test help panel 474
textual configuration report, how to
 print 253
textual configuration reports,
 examples 399
timing-only link 120
title line 55
Tivoli System Automation
 TSA xxvii
token 221
token, esoteric 296
TOP command 56
TRACE command 491
TRACE command and HCM 494
TRACE command via HCD profile 494
trace data set 494
trace data set allocation 509
trace facility 488, 509
trademarks 539
transfer of IODF to another location or
 system 496
transferring partitions 100
transmit a configuration package 49
 problem determination 480
TSA
 Tivoli System Automation xxvii
type of channel path, changing 121
typing over existing data 61

U

UIM 548
 library name 24
 list of installed ones 276
 problem determination 478
 use by HCD 8
unattended export 41
unattended fast-path mode 187
uninstalled ports 176
unit address range 129
UNITNAME statement, esoteric 296
UNITNAME statement, generic 297
unnumbered multiple selection lists 58
update 369
 update access 360
 update CTC connections 95
 updating parts through migration 313
upgrade
 IODF, batch 330
 IODF, dialog 44
usage type 317

user authentication
 export IODF
 sending user ID and PW 42
 using surrogat user ID 42
user interface
 ISPF 531
 TSO/E 531
utility functions, batch 327

V

V5 IODF 40, 45
validated work IODF 225, 548
validation
 during build production IODF 208
 during migration 289
 rules for migration 280
 switch configuration data 165
 what HCD does 3
verify CTC connection 156
verify your definitions 258
verifying configuration
 against local system 228
 against remote system 228
verifying configuration, prerequisites 9
view graphically 258
View I/O Definition File Information
 panel 41
viewing
 activity log 270
 CF control unit and devices 155
 configuration data 73
 coupling facility information 155
 currently accessed IODF 41
 destination channel path
 definition 155
 graphical reports 260
 IODF used for last IPL 38, 230
 logical control units 155
 messages 236
 next IODF processor 215
 object definitions 154
 source channel path definition 155
VIO (virtual input/output) 548
VM
 build HCPRIo input data set,
 batch 338
 build OS configuration data set 226
 considerations 367
 console report, example 421
 device detail report, example 421
 device report, example 420
 load UIMs 24
 migration considerations 281
VM guest, sysplex-wide activate 235
VOLSER, other data sets 22
volume serial number, IODF 33
VSAM data set, initialize 329

W

wait state 472
wildcard processing, example 71
work area 55
work IODF 548
 build with batch 335

work IODF (*continued*)

 format of 31
 how to specify 31
working with
 channel path 108
 consoles 150
 control units 126
 devices 134
 EDTs 80
 esoterics 82
 generics 82
 operating system configuration 77
 partitions 97
 processors 85
 switches 163
world-wide panel names generation 227
write IOCDS 212
write protection, enable and disable 215
WWPN Prediction Tool 227, 549
 FCP SAN configuration template
 file 227

X

XMP processor 549
XMP processors
 term definition 85

Z

z Systems cluster support 11
z/OS discovery and I/O
 Autoconfiguration
 zDAC xxvii
zEDC-Express 103
zooming graphical reports 261



Product Number: 5650-ZOS

Printed in USA

SC34-2669-04

