IBM z/OS HCD & HCM
Newsletter No 29

How to Migrate your Environment to z/VM HCD

How to Migrate your Environment to z/VM HCD

Contents

Chapter 1. Introduction

To manage the I/O configuration with HCD under z/VM, you have the following options:

- You can use the provided REXX utilities to perform z/VM HCD tasks.
- You can use HCM as a graphical user interface to z/VM HCD. Therefore, you must load the HCD IODF into an HCM configuration file. For more information, refer to the *z/OS and z/VM Hardware Configuration Manager User's Guide*.
- You can decide to manage your operating system configuration in addition to the hardware, or you can use HCD just to manage your hardware I/O configuration.

  If you want to use HCD to also control the operating system I/O configuration under z/VM, you have to define an operating system of type VM in the IODF, assign the appropriate devices to the VM operating system, and define the VM consoles.

As prerequisite for managing your I/O configuration with HCD, you need an IODF belonging to z/VM HCD. This document describes the steps how to produce an IODF from an IOCP deck.

The IODF contains the logical I/O configuration definitions that are required for writing an IOCDS, for building the RDEV control blocks at IPL time, and for dynamic hardware and software changes.

An HCM configuration file is always strongly associated to a specific IODF and contains logical data from this IODF as well as additional data on physical I/O configuration aspects. Thus, it contains all data that is necessary to show an I/O configuration diagram from the physical configuration point of view.

If z/OS is used in your environment, you already have an IODF file. It is then recommended to define the I/O configuration under z/OS HCD, if you decided to manage your z/VM configuration using HCD. Thus, the hardware configuration for both the z/OS and z/VM systems can be managed from a central place.

On z/VM, the user ID that is used for HCD/HCM is CBDIODSP. This user ID is the *z/VM HCD TCP/IP dispatcher*. The work and production IODF files are stored on the 191 disk. The z/VM HCD/HCM production files are on the 400 disk. These include the **CBDSxxxx** REXX utilities.

Chapter 2. Migration from an IOCP

It is recommended to save the original IOCP file before preparing it for the migration task.

Before an existing IOCP file can be migrated into an IODF, in most cases, it has to be updated for the proper control unit and device types. The IOCP program, generally, does not validate the UNIT and MODEL keywords on the CNTLUNIT and IODEVICE control statements, whereas HCD does. Therefore, these statements must have unit and model values which pass the HCD validation function.

The **Supported Hardware Report** contains details on which control unit types and device types are supported by HCD and which device types can be connected to which control unit types. You can create a **Supported Hardware Report** using the **CBDSREP** REXX utility or you can produce this report using HCM. A sample **Supported Hardware Report** is presented in *z/VM I/O Configuration*.

For generating an HCM configuration file from the IODF, it is also recommended to prepare the IOCP file by adding physical connection data using extended migration statements into the IOCP file to be migrated. Examples on how to do this are provided in "Step 1. Prepare the IOCP deck for HCD" on page 10. The HCD migrate function will put this data into the IODF, and HCM can use that additional information to produce an appropriate physical configuration diagram.

A native z/VM IOCP deck just contains I/O configuration statements according to the IOCP syntax rules. However, besides the information allowed by the IOCP syntax, HCD introduces an extended migration to allow you to define your complete logical and physical configuration in the IOCP deck. For example, the extended migration allows you to define a switch with its ports, or to define serial numbers and descriptions for devices and control units by manually editing your IOCP input file according to the syntax rules of both the IOCP statements and the HCD extended migration statements. They help you to describe your hardware and infrastructure in your IODF and help HCM to provide a more complete configuration diagram with physical aspects.

The extended migration statements are written as comments into the IOCP, prefixed by one of two tags in column 1:
- *HCD* prefixes extended migration statements, for example for switches and ports.
- *$HCDC$* prefixes extended migration statement keywords, for example for serial numbers and switch port connections.

The extended migration data consists of:

Partition usage types

> HCD validates a logical partition either as usage for an operating system (OS) partition, a coupling facility (CF) partition, or both (CF/OS). For this case, it is recommended to add the partition usage data as comment line following the RESOURCE statement:

> **Example:**

The following RESOURCE statement has added an $HCDC$ tagged comment line that defines logical partitions LP1, LP2 and LP3 with usage type OS and logical partition CF1 with usage type CF.

```
          RESOURCE   PART=((LP1,1),(LP2,2),(LP3,3),(CF1,4))
*$HCDC$        USAGE=(OS,OS,OS,CF)
```

Coupling facility connection information

HCD keeps connection information for the coupling facility in the IODF. This information cannot be determined from the pure IOCP statements. To put that data into the IODF, add the TPATH keyword as an $HCDC$ tagged comment line following a CHPID statement for a coupling facility channel path.

Example:

The following CHPID statement defines coupling facility peer channel path 10. It has added as comment the target and source connection information. The target is CF receiver channel 11 of processor configuration PROC1 using control unit FFFD and devices starting with FFF2 up to FFF8. The source is channel path 10 of processor PROC2 using control unit FFFE and devices starting with FFF9 up to FFFF.

```
          CHPID PATH=(CSS(0,1),10),SHARED,PCHID=100,TYPE=CFP
*$HCDC$        TPATH=((CSS(0),PROC1,11,FFFD,FFF2),              *
               (CSS(0),PROC2,10,FFFE,FFF9))
          CNTLUNIT CUNUMBR=FFFE,PATH=((CSS(0),10),(CSS(1),10)), *
               UNIT=CFP
          IODEVICE ADDRESS=(FFF9,007),CUNUMBR=(FFFE),UNIT=CFP
```

Switch data For each ESCON or FICON directors used in your configuration, pass a SWITCH comment statement.

Example:

The following HCD tagged comment line defines a FICON director 2032 with ID 10 and installed ports 04 to 43. If you have director-to-director (switch-to-switch) connections, then this information needs to be provided as well via the SWPORT comment statement, as shown in the example, where switch 10 has connections from port 06 to switch 11/port 05, from port 07 to switch 11/port 04 and from port 08 to switch 13/port 0A.

```
*$HCD$   SWITCH   SWID=10,ADDRESS=61,DESC='McData ED-6064 z/OS',  *
               PORT=((04,43),(FE,FE)),UNIT=2032,                 *
               SWPORT=((06,11,05),(07,11,04),(08,13,0A))
```

Connection information for ESCON and FICON directors (switch and port data) on CHPID and CNTLUNIT statements

For each channel path connecting to a dynamic switch (CHPID statement with SWITCH keyword) and each control unit that has link addresses defined (CNTLUNIT statement with LINK keyword), add the switch connection data as $HCDC$ tagged comment lines after the CHPID and CNTLUNIT statements.

Example:

CNC channel path 11 is connected to switch 20 port A1, control unit 0400 is connected to switch 20, ports 8B and B7, and to switch 21, ports 9A and BC.

```
                CHPID PATH=(11),TYPE=CNC,SHARED,SWITCH=20
*$HCDC$         SWPORT=((20,A1))
                CNTLUNIT CUNUMBR=0400,UNIT=3990,PATH=(20,30,40,50),        *
                LINK=(8B,B7,9A,BC), UNITADD=((00,256))
*$HCDC$         SWPORT=((20,8B),(20,B7),(21,9A),(21,BC))
```

Control unit serial numbers

Specify the same serial number for all control units that belong to the same physical controller. Thus, HCM will arrange these control units into the same physical controller instance.

Example:

The following control units 9000 to 9300 belong to the same physical controller. HCM will put them into the same physical box if they have the same serial number defined.

```
                CNTLUNIT CUNUMBR=9000,PATH=(14,16,19,1B),LINK=(E0,E4,E8,EC),  *
                CUADD=0,UNITADD=((00,64)),UNIT=2107
*$HCDC$         SERIAL='23425'
                CNTLUNIT CUNUMBR=9100,PATH=(18,15,1A,17),LINK=(E0,E8,E4,EC),  *
                CUADD=1,UNITADD=((00,64)),UNIT=2107
*$HCDC$         SERIAL='23425'
                CNTLUNIT CUNUMBR=9200,PATH=(14,16,19,1B),LINK=(E0,E4,E8,EC),  *
                CUADD=2,UNITADD=((00,64)),UNIT=2107
*$HCDC$         SERIAL='23425'
                CNTLUNIT CUNUMBR=9300,PATH=(18,15,1A,17),LINK=(E0,E8,E4,EC),  *
                CUADD=3,UNITADD=((00,64)),UNIT=2107
*$HCDC$         SERIAL='23425'
```

HCD can recognize point-to-point CTC connections if the involved control units of a CTC connection have the same serial numbers. For more information, refer to the *z/OS and z/VM: Hardware Configuration Manager User's Guide* in the topic *Building Your Configuration*.

Chapter 3. Migrating the OS Configuration

To define the I/O configuration to the software, you can either postpone that step until you have generated the HCM configuration file from the IODF. Then, use HCM to define the VM operating system and assign the devices to this operating system. Or, you can migrate the actual operating system I/O configuration settings in terms of the RDEV control blocks into the IODF.

Migration of the RDEV control blocks can be done only from a z/VM system. For this purpose, you first must generate a file which contains the binary image of the RDEV control blocks of your z/VM system. In a second step, migrate this file into the IODF. (see "Step 4. Migrate your operating system from an RDEV list" on page 15).

You may have to complete the definition afterwards in the IODF for the devices that are off-line and cannot be sensed by z/VM.

Chapter 4. Scenario 1: How to Migrate to HCD in a z/VM Environment without z/OS

In this scenario, we assume that no IODF is existing, so that an IODF must be created on one of the z/VM systems.
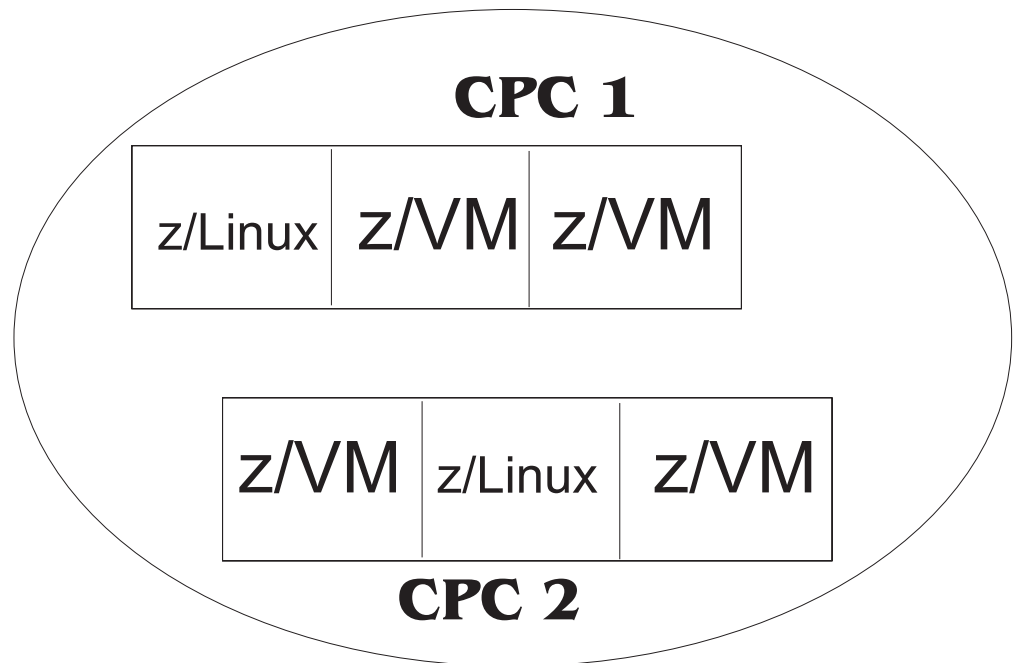
Figure 1. z/VM Environment without z/OS

Here is the list of steps to be performed:
- "Step 1. Prepare the IOCP deck for HCD" on page 10
- "Step 2. Create and initialize an empty work IODF" on page 12
- "Step 3. Migrate the z/VM IOCP into the work IODF" on page 13
- "Step 4. Migrate your operating system from an RDEV list" on page 15
- "Step 5. Check your configuration specification" on page 15
- "Step 6. Build the production IODF" on page 16
- "Step 7. Build the IOCDS" on page 17
- "Step 8. Set up the SYSTEM CONFIG file" on page 17
- "Step 9. Perform a Power on Reset (POR)" on page 18
- "Step 10. Perform an initial program load (IPL)" on page 19
- "Post-Migration configuration management with HCD" on page 22

After having successfully completed the migration to z/VM HCD, you can apply further updates to the IODF using z/VM HCM or z/VM HCD. To enable HCM to draw a usable configuration diagram, observe the recommendations on how to prepare the IOCP input data set (see Chapter 2, "Migration from an IOCP," on page 3) before migrating this data set to the IODF.

Step 1. Prepare the IOCP deck for HCD

For this scenario, we assume as prerequisite that an IOCP deck exists on your z/VM system with the following file name and format:

```
 IOCP file:
 IOCPVM1  IOCP     A1 F         80
```

In the following, a couple of examples are provided of how to apply extended migration statements into the IOCP. Extracts of a sample extended IOCP syntax are shown in Figure 2 on page 12.

The following sections introduce the extended migration statements available for the different configuration objects. In the given examples, the extended migration statements are printed in ***bold/italic*** font. Each example also mentions where HCM lets you benefit from the extended migration statements.

Migration statements for processors

Extended migration statements for processors may contain a description of the processor and its network name and CPC name.

Table 1. Extended migration statements: Processors

Statement(s)	Sample

SNAADDR	`          ID    MSG2='IODF17 PRODIODF A - 2009-05-19 20:05',`
	`                SYSTEM=(2094,1)`
	`*$HCDC$          `***`SNAADDR=(IBM390PS,POL1)`***

Where visible in HCM:
- **Processor** dialog

Migration statements for partitions

Extended migration statements for partitions may describe the usage (OS, CF, or CF/OS) of the partitions of a processor or a channel subsystem.

Table 2. Extended migration statements: Partitions

Statement(s)	Sample

USAGE	`          RESOURCE PARTITION=((CSS(0),(HCDVM1,1),(HCDVM2,2),(HCDVM3,3),(*`
	`                HCDVM4,4),(HCDVM5,5),(HCDVM6,6),(HCDVM7,7),(HCDVM8,8),(H*`
	`                CDVM9,9),(HCDVM10,A),(*,B),(*,C),(*,D),(*,E),(*,F)),(CSS*`
	`                (1),(ZLINUX1,1),(ZLINUX2,2),(ZLINUX3,3),(ZLINUX4,4),(ZLI*`
	`                NUX5,5),(ZLINUX6,6),(ZLINUX7,7),(ZLINUX8,8),(ZLINUX9,9),*`
	`                (ZLINUX10,A),(*,B),(*,C),(*,D),(*,E),(*,F))),          *`
	`                MAXDEV=((CSS(0),20000,1000),(CSS(1),20000,1000))`
	`*$HCDC$          `***`USAGE=(CF/OS,OS,OS,OS,OS,OS,OS,OS,OS,CF/OS,OS,OS,OS,OS,C*`***
	`F/OS,CF/OS,OS,OS,OS,OS,OS,OS,OS,OS,OS,OS,OS,OS,OS,OS)`

Where visible in HCM:
- **Partitions** dialog

Migration statements for CHPIDs

Extended migration statements for CHPIDs may specify the switchport to which the CHPID connects, and a description of the CHPID.

Table 3. Extended migration statements: CHPIDs

Statement(s)	Sample
| SWPORT | ```
          CHPID PATH=(CSS(0),10),SHARED,                                    *
                PARTITION=((P8),(POLLP01,POLLP02,POLLP03,POLLP04,POLLP05*
                ,POLLP06,POLLP07,POLLP10)),SWITCH=02,PCHID=160,TYPE=CNC
*$HCDC$        SWPORT=((02,27))
``` |

Where visible in HCM:

- in the **CHPID** dialog
- in the **Edit CHPID** dialog
- in the HCM configuration diagram: visible as connection between CHPID 10 and switch port 27 of switch 02

Migration statements for control units

Extended migration statements for control units may contain a description, a serial number and the switches and ports, where they are connected

Table 4. Extended migration statements: Control units

Statement(s)	Sample
| SERIAL<br>SWPORT | ```
          CNTLUNIT CUNUMBR=0200,PATH=((CSS(0),36,38),(CSS(1),36,38)),    *
                   UNITADD=((00,080)),LINK=((CSS(0),1F,9F),(CSS(1),1F,9F)),*
                   CUADD=3,UNIT=1750
*$HCDC$           SERIAL='3233       '
*$HCDC$           SWPORT=((17,1F),(17,9F))
``` |

Where visible in HCM:

- in the **Locate One or More Control Units** list
- in the **Control Unit List** dialog
- in the HCM configuration diagram

Migration statements for switches

Extended migration statements for switches may contain the switch ID, a description, information about the type and model and the installed ports.

Table 5. Extended migration statements: Switches

Statement(s)	Sample
| SWITCH | ```
*$HCD$    SWITCH    SWID=02,PORT=((0C,FB),(FE,FE)),MODEL=5,UNIT=9032
*$HCD$    SWITCH    SWID=17,ADDRESS=68,PORT=((00,1F),(80,9F)),UNIT=FCS
*$HCD$    SWITCH    SWID=10,ADDRESS=61,PORT=((04,43),(FE,FE)),UNIT=2032, *
                    SWPORT=((06,17,05),(07,17,04))
``` |

Where visible in HCM:

- in the **Locate One or More Switches** list
- in the **Edit Switch** dialog

Example of an IOCP deck using extended migration statements

```
            ID    MSG2='IODF17 PRODIODF A - 2009-05-19 20:05',          *
                  SYSTEM=(2094,1)
 *$HCDC$          SNAADDR=(IBM390PS,POL1)
            RESOURCE PARTITION=((CSS(0),(HCDVM1,1),(HCDVM2,2),(HCDVM3,3),(*
                  HCDVM4,4),(HCDVM5,5),(HCDVM6,6),(HCDVM7,7),(HCDVM8,8),(H*
                  CDVM9,9),(HCDVM10,A),(*,B),(*,C),(*,D),(*,E),(*,F)),(CSS*
                  (1),(ZLINUX1,1),(ZLINUX2,2),(ZLINUX3,3),(ZLINUX4,4),(ZLI*
                  NUX5,5),(ZLINUX6,6),(ZLINUX7,7),(ZLINUX8,8),(ZLINUX9,9),*
                  (ZLINUX10,A),(*,B),(*,C),(*,D),(*,E),(*,F))),
                  MAXDEV=((CSS(0),20000,1000),(CSS(1),20000,1000))
 *$HCDC$          USAGE=(CF/OS,OS,OS,OS,OS,OS,OS,OS,OS,CF/OS,OS,OS,OS,OS,C*
                  F/OS,CF/OS,OS,OS,OS,OS,OS,OS,OS,OS,OS,OS,OS,OS,OS,OS)
            CHPID PATH=(CSS(0,1),0C),SHARED,                             *
                  NOTPART=((CSS(0),(HCDVM1),(HCDVM2)),(CSS(1),            *
                  (ZLINUX1),(=))),                                       *
                  PCHID=110,TYPE=OSC
            CHPID PATH=(CSS(0),10),SHARED,                              *
                  PARTITION=((HCDVM2,HCDVM3),(HCDVM9,HCDVM10)),          *
                  SWITCH=02,PCHID=160,TYPE=CNC                          *
 *$HCDC$          SWPORT=((02,27))
 ...
            CNTLUNIT CUNUMBR=0003,PATH=((CSS(0),0C),(CSS(1),0C)),UNIT=OSC
            IODEVICE ADDRESS=(090,016),MODEL=X,UNITADD=00,CUNUMBR=(0003), *
                  UNIT=3270
            IODEVICE ADDRESS=(1090,004),MODEL=X,UNITADD=10,CUNUMBR=(0003),*
                  UNIT=3270
            CNTLUNIT CUNUMBR=0010,PATH=((CSS(0),32)),UNITADD=((00,008)),  *
                  CUADD=3,UNIT=FCTC
            IODEVICE ADDRESS=(010,008),UNITADD=00,CUNUMBR=(0010),STADET=Y,*
                  UNIT=FCTC
            CNTLUNIT CUNUMBR=0200,PATH=((CSS(0),36,38),(CSS(1),36,38)),  *
                  UNITADD=((00,080)),LINK=((CSS(0),1F,9F),(CSS(1),1F,9F)),*
                  CUADD=3,UNIT=1750
 *$HCDC$          SERIAL='3233        '
 *$HCDC$          SWPORT=((17,1F),(17,9F))
            IODEVICE ADDRESS=(200,020),CUNUMBR=(0200),STADET=Y,          *
                  NOTPART=((CSS(0),P8,HCDVM),(CSS(1),D2)),UNIT=3390B
            IODEVICE ADDRESS=(214,060),CUNUMBR=(0200),STADET=Y,          *
                  NOTPART=((CSS(0),P8,HCDVM),(CSS(1),D2)),UNIT=3390A
 ...
 *$HCD$   SWITCH   SWID=10,ADDRESS=61,                                  *
                  PORT=((04,43),(FE,FE)),UNIT=2032,                     *
                  SWPORT=((06,11,05),(07,11,04),(08,13,0A))
 *$HCD$   SWITCH   SWID=11,ADDRESS=62,                                  *
                  PORT=((04,83),(FE,FE)),UNIT=2032,                     *
                  SWPORT=((04,10,07),(05,10,06))
```

Figure 2. Extracts from an IOCP with extended migration statements

---

Step 2. Create and initialize an empty work IODF

Create an empty work IODF into which the IOCP deck prepared in Step 1 on page 10 will be migrated. For this purpose, use the **CBDSIODF** REXX utility (see "Create and Initialize an IODF (CBDSIODF)" in *z/VM I/O Configuration*).

Example:

```
cbdsiodf iodf71 workiodf a 1024
```

This command creates an IODF file with the following properties:

```
filename:   filetype:   filemode:   format:  LRECL:   records:  blocks:
IODF71      WORKIODF    A6          F        4096     1024      1024
```

Step 3. Migrate the z/VM IOCP into the work IODF

In this step, we migrate the IOCP input file prepared in Step 1 into the work IODF created in Step 2. We do this by invoking the REXX utility **CBDSMIGR** to migrate the IOCP file called IOCPVM1 IOCP A into the IODF71 WORKIODF A under the new processor configuration POL1 of type 2094-S07 in LPAR mode, using support level H070331.

If different support levels are available for a processor type, you can specify the appropriate one. Otherwise, the highest support level is taken as a default. You can retrieve available support levels from the **Supported Hardware Report**.

Example:
```
cbdsmigr iodf71 workiodf a iocpvm1 iocp a i pol1 2094-S07 lpar h070331
```

For more information about the migration function, see topic *How to Migrate Existing Input Files* in *z/VM I/O Configuration*.

Prerequisite: To generate the additional keywords HCD and $HCDC$ during IOCP file build, specify the following entry in the HCD PROFILE (which resides on the CBDIODSP user ID's A-disk):
```
MIGRATE_EXTENDED = YES
```

After the migration, file CBDSMIGR MESSAGES A informs you about a successful completion: `Return Code was    0`

How to debug a migration problem

Depending on errors in the IOCP input file, HCD may create the following files:
```
CBDSMIGR MESSAGES A
CBDSMIGR MSGLOG    A
CBDSMIGR LISTING   A
```

If HCD creates an assembler listing, it may show more error details. Correct the error and run CBDSMIGR again. Now HCD just creates the MESSAGES and MSGLOG files.

Let us assume, you coded an IODEVICE statement for a 3270 device without a required `MODEL=X` specification:

incorrect IODEVICE statement:
```
        CNTLUNIT CUNUMBR=0003,PATH=((CSS(0),0C),(CSS(1),0C)),UNIT=OSC
        IODEVICE ADDRESS=(090,016),UNITADD=00,CUNUMBR=(0003),        *
            UNIT=3270
```

same IODEVICE statement corrected:
```
        CNTLUNIT CUNUMBR=0003,PATH=((CSS(0),0C),(CSS(1),0C)),UNIT=OSC
        IODEVICE ADDRESS=(090,016),MODEL=X,UNITADD=00,CUNUMBR=(0003), *
            UNIT=3270
```

In such a case, HCD writes the following message into the related MESSAGES file:

```
HCD                                    Time: 14:06 Date: 09-07-09  Page:    1
z/VM 5.4.0 HCD

(1) FILE=IOCPVM1 IOCP A  1


Statement Orig Sev  Msgid    Message Text

     54   (1)    E  CBDA297I  Control unit 0003 of type OSC cannot attach device 0090 of type 3270.
                I  CBDA516I  No output written to IODF. VALIDATE processing forced due to errors.


Total Messages  Terminating     Error     Warning  Informational
       2                          1                      1

Return Code was   8  2
```

Figure 3. CBDSMIGR MESSAGES A1

The message points to an error in file IOCPVM1 IOCP A **1** and shows the
assembler processing return code **2** .

How to handle migration errors due to HCD validation checks

As HCD performs strict validation checks, you may receive error messages when
migrating your IOCP deck, that had been tolerated by z/VM.

With message *CBDA155I* as an example, this section shows you how to find the
erroneous statement and how to correct the error.

Suppose, in the CBDSMIGR MESSAGES file, z/VM HCD reports error *CBDA155I*
3 in statement 179 **2** of IOCP deck IOCPVM1 IOCP A **1** :

```
HCD                                    Time: 11:10 Date: 09-07-28  Page:    1
z/VM 5.4.0 HCD

(1) FILE=IOCPVM1 IOCP A  1

Statement Orig Sev  Msgid     Message Text

   179 2 (1)    E  CBDA155I 3 Too many channel paths of type OSD, ... defined.
                              Maximum is 48 for processor POL1.
```

In file CBDSMIGR LISTING, you can retrieve the IOCP statement 179 by searching
for the string: *179 IS THE PREVIOUS IOCP/MVSCP STATEMENT NUMBER.*

The statement preceding this text line is the IOCP statement in error:
```
 CHPID PCHID=611,PATH=(CSS(0,1,2,3),93),TYPE=OSD,SHARED
*, 179 IS THE PREVIOUS IOCP/MVSCP STATEMENT NUMBER
```

Now look for a user response to this message in *z/OS and z/VM Hardware
Configuration Definition Messages*. In our case you find the following
recommendations:

User response: You may do one of the following:
- Reduce the number of channel paths of the given type(s).
- Choose a different processor type-model or support level if available.

Step 4. Migrate your operating system from an RDEV list

Note: If you do not want to use HCD to control the software (OS) configuration, skip this step. Otherwise, you can use two z/VM HCD REXX utilities **CBDSRDEV** and **CBDSMIGR** to perform this task.

Step 4a: Create a RDEVLIST from your active VM system

First, by using utility **CBDSRDEV**, you generate an image of the RDEV control blocks from the running system. In other words, you create a file containing the current CP I/O configuration extracted from the active VM system (or more exactly, from the RDEV control blocks).

CBDSRDEV invocation syntax:

```
CBDSRDEV [outfile] [(REPlace]
```

outfile With this optional parameter, you can specify an output file for the I/O configuration. The default is CBDMCONF RDEVLIST A.

Example:

```
cbdsrdev vmconf rdevlist a
```

You will see the following console message:

```
********************************************************************************
    Successful completion. File   VMCONF RDEVLIST                  A
    created.
********************************************************************************
```

Step 4b: Migrate the RDEVLIST into your work IODF

To migrate the I/O configuration retrieved in step 4a into your IODF (file IODF71 WORKIODF A) under the OS configuration name REALVM, you again use the **CBDSMIGR** utility:

```
CBDSMIGR utility:
  cbdsmigr iodf71 workiodf a vmconf rdevlist a r realvm
```

At this point it is important to check the created files CBDSMIGR MESSAGES and CBDSMIGR MSGLOG. **CBDSRDEV** gets the device information from an RDEV control block. HCD matches the devices defined in the work IODF against the ones found in the CBDSRDEV output. If a device defined in the IODF is not online/available, then no control block entry has been found, and CBDSMIGR MESSAGES shows a corresponding warning message.

Step 5. Check your configuration specification

Now you have prepared a valid work IODF. At this point, it is recommended to verify the configuration and check, whether all is defined as you intended. For this purpose, you can use REXX utility **CBDSREP** to generate a variety of reports of your work IODF, for example, a *Channel Subsystem (CSS) Report* or a *VM Device Report*. With a parameter, you specify which type of report you want. How to invoke this utility is described in topic *Print Configuration Reports (CBDSREP)* of *z/VM I/O Configuration*. Check the reports. If you discover discrepancies, you can

manually update your original IOCP deck with your corrections. Remigrate the changed IOCP into your work IODF (**CBDSMIGR**). Now use parameter *IP* for partial migration instead of *I* in "Step 3. Migrate the z/VM IOCP into the work IODF" on page 13.

If you performed the optional "Step 4. Migrate your operating system from an RDEV list" on page 15, it is recommended to compare the devices defined for the processor and the operating system. Run the REXX utility **CBDSCONF** to produce an output file from your work IODF containing I/O configuration statements for your operating system configuration.

The following REXX utility invocation produces the operating configuration statements for the operating system called REALVM:

```
cbdsconf iodf71 workiodf a realvm deck a o realvm
```

In the excerpt from output file REALVM DECK A shown in Figure 4, you see that, for example, device 1100 is defined for the operating system:

```
...
IOCONFIG ID=00,NAME=REALVM,TYPE=VM
RDEVICE DEVNO=(1100,1),DEVTYPE=3215,OFFLINE=NO,UIRATE=DEFAULT
...
```

Figure 4. Excerpt from REALVM DECK A

Let us assume, there is no RDEVICE statement for device 1000 contained in file REALVM DECK A, but this device is defined for your processor POL1 configuration. If you also want to define this device for your operating system, you must define the missing device manually into your deck REALVM DECK A by adding the appropriate RDEVICE statement for device 1000.

Also, there may have been devices that had not been online at the time you ran the **CBDSRDEV** utility. Such devices are not defined in the current work IODF, because offline devices cannot be sensed by z/VM. You can now add these definitions to the OS configuration (REALVM). Therefore, you again add an appropriate RDEVICE statement for the missing device into your REALVM DECK A output file.

Now remigrate your completed REALVM DECK into your work IODF IODF71 WORKIODF A using partial migration ("Step 3. Migrate the z/VM IOCP into the work IODF" on page 13):

```
cbdsmigr iodf71 workiodf a realvm deck a op realvm
```

Step 6. Build the production IODF

Now that you have created a correct and complete work IODF, it is recommended to save a backup of this file. Then use the REXX utility **CBDSPROD** to build the production IODF.

Example:
```
cbdsprod iodf71
```

This is the shortest possible invocation of the **CBDSPROD** utility. It creates a production IODF called IODF71 PRODIODF A. If errors occur, these are written into the message log CBDSPROD MSGLOG A.

Check the severity of the error messages in the message log in column *Sev*:

- Informational messages are denoted by a letter **I**.
- Warnings are denoted by a letter **W**.
- Error messages are denoted by a letter **E**.

Check the messages and the user responses in *z/OS and z/VM Hardware Configuration Definition Messages*. You must correct the error messages, because otherwise, HCD cannot build a production IODF. For the information messages and the warnings, you can decide whether you want to change the IODF definitions. HCD builds a production IODF despite these messages, however, using the resulting configuration later may lead to undesired or unexpected results.

Step 7. Build the IOCDS

With a correct production IODF, use the REXX utility **CBDSIOCP** to perform an IOCDS download to a specified IOCDS slot of a processor specified in the production IODF. Assume, we have one processor POL1 defined in the IODF, and we want to download the IOCDS to the support element into slot A0. The related command would be like follows:

```
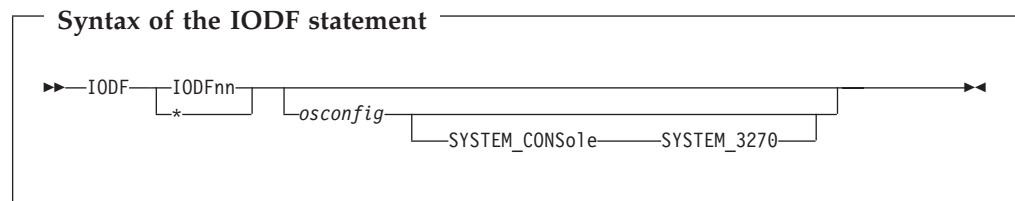cbdsiocp iodf71 prodiodf a pol1 i a0
```

Step 8. Set up the SYSTEM CONFIG file

The IODF statement in the SYSTEM CONFIG file indicates that HCD will be used to control the I/O hardware and/or software configuration.

```
┌─ Syntax of the IODF statement ──────────────────────────┐
│                                                          │
│  ►►─IODF─┬─IODFnn─┬─┬──────────┬────────────────────────►◄ │
│          └─*──────┘ └─osconfig─┬──────────────────────┐   │
│                                └─SYSTEM_CONSole──SYSTEM_3270─┘ │
│                                                          │
└──────────────────────────────────────────────────────────┘
```

In the SYSTEM CONFIG file of z/VM, you specify which IODF and which z/VM configuration is to be used for IPL. The parameters in the IODF statement have the following meanings:

IODFnn

The file name of the production IODF. You must specify the name of a file of file type PRODIODF, and the file must be located on the system parameter disk (SYSPARM). If the correct IODF is not located on the system parameter disk, you may run into a 'disable wait problem' (see "Disable Wait problem during IPL" on page 20).

*
An * specifies that the filename of the production IODF to be used is the one currently stored in the hardware configuration token in the HSA. The matching IODF must be located on the system parameter disk (SYSPARM) at the system IPL time. In our example filelist shown hereafter, the IODF name is *IODF71*.

```
Filename Filetype Fm Format Lrecl    Records

SYSTEM   CONFIG   B1 F        80        272
IODF19   PRODIODF B6 F      4096        184
SNAZ     LOGO     B1 F        78         16
PRINTZ   LOGO     B1 F        78         16
LOCALZ   LOGO     B1 F        78         16
LDEVZ    LOGO     B1 F        78         16
```

```
IODF71    PRODIODF B6 F       4096        7000
CPLOAD    MODULE   B1 V      65535         149
CPLOLD    MODULE   B1 V      65535         149
```

The following specifications are only applicable if HCD also controls the software I/O configuration.

osconfig

specifies the OS configuration ID of the VM I/O configuration that is defined in the IODF. This OS configuration should be controlled with HCD. If specified, this configuration is used to build the software view of the I/O configuration, and the statements that are normally used in the system configuration file to build this view are ignored.

SYSTEM_CONSole

specifies that the Operating System Messages panel on the IBM Hardware Management Console (HMC) can serve as a system operator console.

SYSTEM_3270

specifies that the integrated 3270 console on the HMC can serve as a system operator console.

Example of an IODF statement:

```
IODF IODF71 REALVM
```

Step 9. Perform a Power on Reset (POR)

On the Hardware Management Console (HMC), go to *Customize Activation Profiles*. Then, if you want to provide for dynamic configuration changes, be sure to adapt the *Reset Profile*. You enable dynamic changes by clicking on *Allow dynamic changes ...* (see Figure 5 on page 19).

Now you can perform a POR with slot A0.

Figure 5. Customize Activation Profiles

Step 10. Perform an initial program load (IPL)

IPL your system. After IPL, type the q hcd command on your z/VM console to see the active IODF. In the example response, you can see that only dynamic HW changes are enabled and that HCD is not controlling the software configuration, because no osconfig parameter had been specified in the SYSTEM CONFIG file.

```
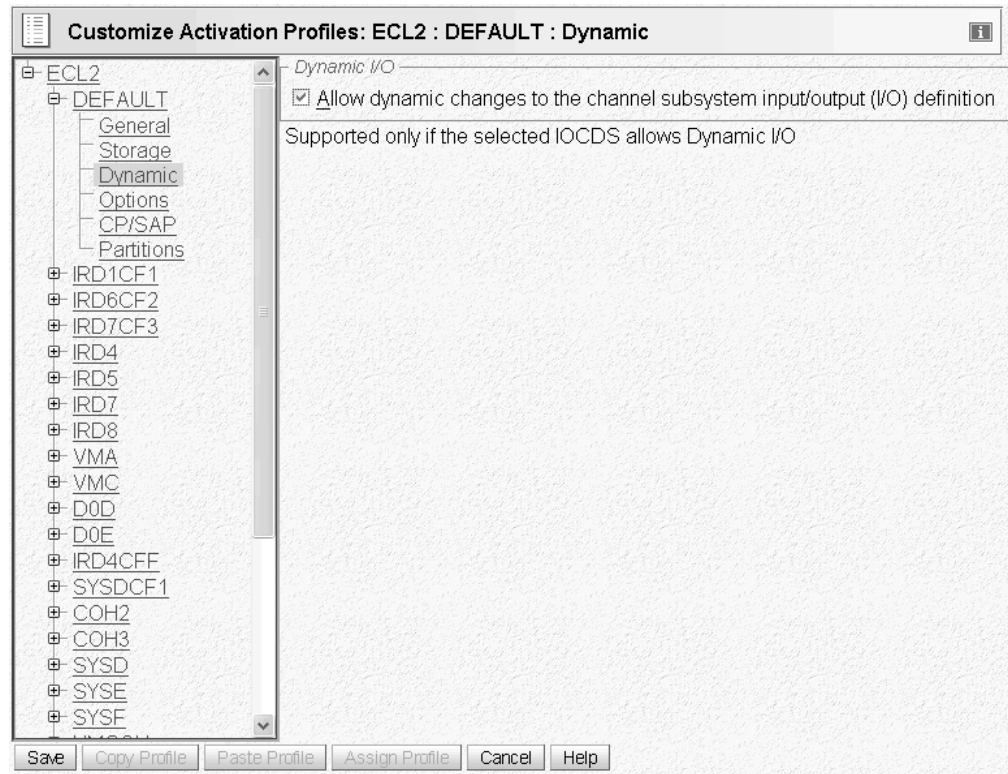HCD is currently active: IODF = IODF71 PRODIODF
HCD is enabled for dynamic hardware changes
HCD is not controlling the software configuration
HCD recovery is not currently required
Ready; T=0.01/0.01 09:53:09
```

You can enter the CP QUERY TOKEN command to see the current or target configuration tokens:

```
q token current

The current HCD configuration token is:
POL1....m.o.................10-03-1007:50:15SYS4   IODF71
Ready;

q token target

The target HCD configuration token is:
POL1....m.o.................10-03-1007:50:15SYS4   IODF71
Ready;
```

Note: The HCD token does not contain VM-TOKEN. It contains the name of the processor (POL1) and of the production IODF (IODF71).

Go to the SE (Support Element), then go to Input/Output (I/O Configuration). On this screen it shows the active IOCDS, Then go to the tab, view Dynamic. On the *Dynamic Information* screen on the SE console (Figure 6) you can check the configuration *Data set token* and the *Hardware system area token*. If HCD should be enabled for dynamic activation changes, both must be identical.

Figure 6. Information display from the support element

> **Note:**
>
> Now your system is up and running according to your applied setup. As you provided for software and hardware changes, your system is also prepared for dynamic configuration changes. For more information, refer to topic *Activate a Configuration Dynamically* in *z/VM I/O Configuration*.

Disable Wait problem during IPL

A Disable Wait problem can occur if the IODF to be loaded is not on the system parameter disk.

Disable Wait 9043

Problem Analysis - POL1

System name:	POL1
Date:	May 19, 2009
Time:	6:58:34 AM

Problem Description

Central processor (CP) 0 in partition HCDVM, entered disabled wait state.
The disabled wait program status word (PSW) is
0002000000000000000000000000(9043)
Central storage bytes 0-7 are: 000c000080039030060002006000020.

Corrective Actions

Refer to your operating system console or documentation for recovery procedures.

Impact of Repair

| Request Service... | No Service | Display Sense Data | Delete | Cancel | Help |

Figure 7. Disable Wait 9043

In this case, message HCP9043W provides the appropriate user response (see also in z/VM CP Messages and Codes):

HCP9043W Unable to process IODF file

Explanation: An IODF statement in the system configuration file was being processed during system initialization and one of the following errors was encountered:
- The specified IODF name is not valid.
- The IODF could not be opened.
- The IODF could not be read.
- The data in the IODF was not valid.
- An osconfig name was specified, but no matching OSR record was found in the IODF.

System Action: The system enters a disabled wait state (wait-state code = 9043).

User Response: Verify the validity of the IODF specified on the IODF system configuration file statement and re-IPL, or re-IPL using the NOHCD parameter on the SAPL screen to bypass the IODF system configuration file statement.

Step 11. Load the configuration into HCM

Now you are ready to run your normal configuration business. To manage your configuration with HCM, at this point in time, you can load your I/O configuration into HCM. Then you can explore your configuration by viewing the HCM configuration diagram and you can use the graphical HCM dialogs to apply

logical and physical changes to your configuration. The logical changes are automatically propagated to the associated IODF on the host, for example, when adding new processors or devices. For detailed information refer to the *z/OS and z/VM Hardware Configuration Manager User's Guide*.

The normal workflow for applying configuration changes with HCM is like follows:
1. Create a new work IODF from your production IODF.
2. Apply your updates.
3. Check the results in the diagram or using the HCM functions to produce HCD textual reports.
4. Build the new production IODF.
5. With the production IODF, use the REXX utility **CBDSIOCP** to perform an IOCDS download to the processor.
6. Dynamically activate your changes as described in "Post-Migration configuration management with HCD."

Post-Migration configuration management with HCD

Using HCD, you can now activate dynamic hardware and software changes in a running system configuration without the need to POR or IPL the system.

Perform the following steps:
1. Check, whether your system setup meets the prerequisites for performing dynamic hardware activation changes:

   Have a look at the tokens displayed in the screen shown in Figure 6 on page 20: the *Data set token* and the *Hardware system area token* must be identical.
2. Create a new work IODF based on your current production IODF:

   If you are using HCM:

   Select action *File ---> Make Work*

   If you are not using HCM:

   a. Use the REXX utility **CBDSCONF** to produce a current IOCP deck from your production IODF. Specify the parameters for this utility depending on whether you want to make changes to the processor configuration or to the operating system.

   b. Apply your changes to the generated IOCP deck. Then use REXX utility **CBDSMIGR** with parameter *IP* for partial migration to migrate the IOCP changes into your work IODF.
3. If you work with HCM you can now check the results in the configuration diagram or you can use the HCM functions to produce HCD textual reports.
4. Build the production IODF as described in "Step 6. Build the production IODF" on page 16.
5. Use the **CBDSACT** REXX utility to test your specified changes without a real activation:

   ```
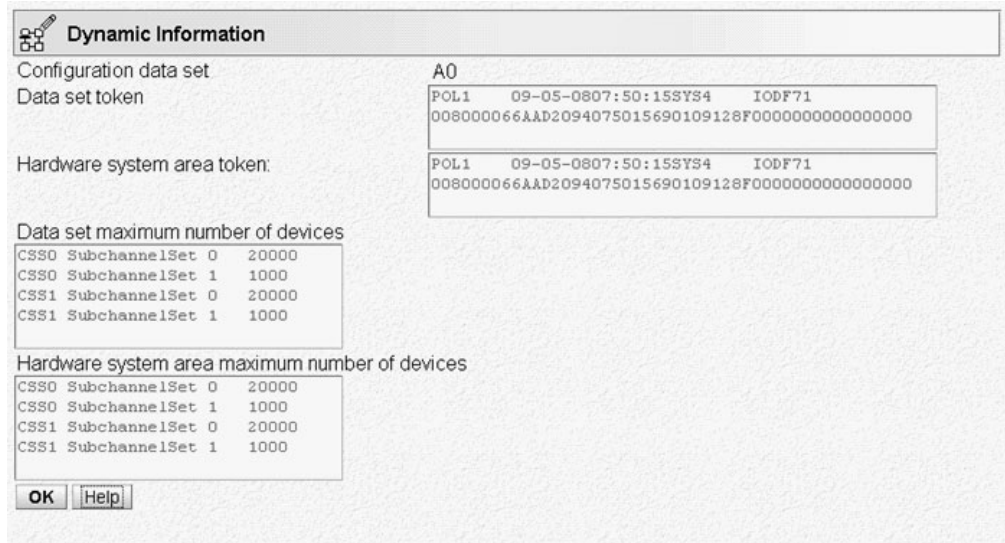   CBDSACT IODF71 * * * (TEST
   ```

   Then look for the result in file CBDSACT MSGLOG:

   ```
    z/VM 5.4.0 HCD                        Time: 09:00 Date: 2010-02-03  Page:   1


   IODF71 PRODIODF A
   Sev  Msgid    Message Text
    I  CBDA126I  ACTIVATE IODF=71,CFID=ECL2,TEST command was accepted.
    I  CBDD798I  The currently active IODF is IODF71 PRODIODF, the active
                 configuration is ECL2.
                 The HSA token is POL1 10-03-10  07:50:15 SYS4 IODF71.
   ```

```
I  CBDA781I  Your system configuration provides full dynamic
              reconfiguration capability.
I  CBDA887I  Following devices are to be added to processor POL1:
              2.2440-2.245F,2.0050-2.006F
I  CBDA884I  Following control units are to be added to processor POL1:
              2.2440,2.0050
I  CBDA881I  Following channel paths are to be added to processor POL1:
              2.16,2.D7
I  CBDD800I  All change requests were successfully executed.
Total Messages  Terminating     Error     Warning  Informational
         7                                               7
Return Code was   0
```

6. If everything is OK as desired, you need to issue the command again without the TEST option to really activate your changes.

   If you control your z/VM software I/O configuration via HCD and you run multiple z/VM systems on *CPC2*, then your dynamic activation task consists of the following steps:

   - Perform a software only change in all z/VM CPs except one.
   - ON the remaining z/VM system, perform a software and hardware change.

7. Send the PRODIODF file to the PARM disk (CF1) on the MAINT user ID. If needed, update the IODF statement in the SYSTEM CONFIG file. Keep a copy of the PRODIODF file that was written out to the IOCDS previously.

8. Use the REXX utility **CBDSIOCP** to perform an IOCDS download. This step is equivalent to "Step 7. Build the IOCDS" on page 17. There is no need for another POR or an IPL of the system.

Note:

Now your system is again up and running with your applied changes and continues to be prepared for further dynamic configuration changes.

Chapter 5. Scenario 2: How to Migrate in a Mixed z/VM and z/OS Environment

A mixed z/VM and z/OS environment designates a customer installation where z/OS is running, but at least one CPC is running completely without z/OS, but with at least one z/VM system (*CPC2* in our example from Figure 8). In such an environment, it is recommended to manage the z/VM system(s) running on this z/OS-free CPC also with z/OS HCD.

In this scenario, we start with an existing work IODF on one of the z/OS systems, containing just the *CPC1* configuration which we want to expand and use as the IODF for the complete environment. Therefore, we want to integrate the z/VM configurations from *CPC2* that does not run a z/OS system, into this IODF.

Figure 8. Mixed z/VM and z/OS Environment

In our scenario, we want to migrate the configuration of *CPC2* which is available as IOCP deck into the IODF located on *CPC1*.

Here is the list of steps to be performed:
- "Step 1. Migrate the z/VM IOCP into the work IODF" on page 26
- "Step 2. Check your configuration specification" on page 26
- "Step 3. Integrate your operating system(s) into the IODF" on page 26
- "Step 4. Building the production IODF" on page 26
- "Step 5. Building the IOCDS" on page 27
- "Step 6. Set up the SYSTEM CONFIG file" on page 27
- "Step 7. POR and IPL" on page 27
- "Post-Migration configuration management with HCD" on page 27

Step 1. Migrate the z/VM IOCP into the work IODF

We assume, that "Step 1. Prepare the IOCP deck for HCD" on page 10 has already been performed on the z/VM system. Therefore, you can now send the IOCP deck from *CPC2* to the z/OS system where the work IODF is located.

The current step 1 in our scenario is functionally equivalent to step 3 in scenario 1. However, in contrast to scenario 1, you can now use z/OS HCD to migrate this IOCP input file into the work IODF:

1. From the Primary Task Selection panel, select *5. Migrate configuration data*.
2. Then select *1. Migrate IOCP/OS data* from the **Migrate Configuration Data** task menu. This invokes the *Migrate IOCP / MVSCP / HCPRIO Data* dialog. See the *z/OS HCD User's Guide* or the online help for detailed information on how to use this dialog.

The IODF accessed by the HCD dialog on the *CPC1* z/OS system must be the work IODF currently containing the *CPC1* configuration. This is the target IODF for the migration of the *CPC2* configuration.

For information on how to transfer files between z/VM and z/OS, see Chapter 6, "How to exchange IODF or IOCP files between z/VM and z/OS," on page 29.

Step 2. Check your configuration specification

At this point, you may again verify your changes with the help of the various HCD reports as described in "Step 5. Check your configuration specification" on page 15. However, in the current scenario, you can use the HCD dialog to produce the reports. From the Primary Task Selection, select path 3.1:

1. *3. Print or compare configuration data*
2. *1. Print configuration reports*

Step 3. Integrate your operating system(s) into the IODF

This is an optional step and is only required if you want to use HCD to control the software (OS) configuration. In this case, perform the following steps:

1. From the Primary Task Selection Panel, select *1. Define, modify, or view configuration data*.
2. Then select *1. Operating system configurations*.
3. In the *Operating System Configuration List* define a new OS configuration of type *VM*. For more information, refer to the HCM User's Guide.
4. Now you can manually assign the required defined devices to the new operating system, starting from the *I/O Device List* as described in the HCD User's Guide.

   Another possibility is to send the IODF to z/VM, then run the **CBDSRDEV** and **CBDSMIGR** utilities as described in "Step 4. Migrate your operating system from an RDEV list" on page 15. Then send the updated IODF back to the z/OS system.

Step 4. Building the production IODF

To generate a production IODF, select path 2.1 from the Primary Task Selection:

1. *2. Activate or process configuration data*
2. *1. Build production I/O definition file*

As the production IODF needs to be present on the system you want to manage, now transfer the production IODF to your z/VM system.

Step 5. Building the IOCDS

For this step, you have two alternatives:

Alternative 1:

If **CPC2** is managed via the Hardware Management Console, you can use path 2.11 of the z/OS HCD dialog to download the IOCP deck:

1. *2. Activate or process configuration data*
2. *11. Build and manage S/390 microprocessor IOCDSs and IPL attributes*
3. From the *S/390 Microprocessor Cluster List*, select **CPC2**.
4. Select *1. Work with IOCDSs*.
5. From the *IOCDS List*, use the '/' marker to select an unused IOCDS slot.
6. This brings up a selection of *Actions on selected IOCDSs*. Select *1. Update IOCDS*. In the upcoming *Build IOCDS* dialog, you do not need to perform any updates, but pressing PF3 just writes your IOCDS into the selected IOCDS slot.

   If you now select *2. Switch IOCDS*, this will update the IOCDS with all changes done to the IODF and will cause the system to automatically use this IOCDS for the next performed POR.

Alternative 2:

You perform the IOCDS download on the z/VM system on **CPC2**. This step is equivalent to "Step 7. Build the IOCDS" on page 17.

After this step, you can manage your complete environment (**CPC1** and **CPC2**) using the z/OS HCD dialog or using the HCM connected to **CPC1**.

Step 6. Set up the SYSTEM CONFIG file

This step is equivalent to "Step 8. Set up the SYSTEM CONFIG file" on page 17 in Scenario 1.

Step 7. POR and IPL

The remaining actions are equivalent to the steps in Chapter 4, "Scenario 1: How to Migrate to HCD in a z/VM Environment without z/OS," on page 9:

- "Step 9. Perform a Power on Reset (POR)" on page 18
- "Step 10. Perform an initial program load (IPL)" on page 19

Post-Migration configuration management with HCD

We assume, you now want to continue to manage your complete environment from a central point of control, using HCD on the z/OS system. This topic describes how to apply configuration changes, also dynamic ones, after you have completed the migration of the z/VM IOCP(s) into the central z/OS IODF of your environment. Basically, you apply your changes on z/OS and send the updated production IODF to z/VM.

1. Use the HCD ISPF dialog to create a new work IODF based on your current production IODF as described in the *z/OS HCD User's Guide*. Continue to use the HCD dialog to apply your configuration changes to the work IODF.

2. Build the production IODF under z/OS HCD: From the Primary Task Selection Panel, select *2. Activate or process configuration data*, then select *1. Build production I/O definition file*.

3. Transfer the production IODF to z/VM. To receive it on z/VM, use the `CMS RECEIVE` command or the **CBDSIMP** REXX utility.

4. Activate the new configuration using the **CBDSACT** REXX utility with the TEST option first and verify these are the changes you want. Then issue the **CBDSACT** REXX utility without the TEST option to perform the dynamic I/O changes.

5. Update either the SYSTEM CONFIG file and the PARM disk or both. Logon to the user ID MAINT and copy the PRODIODF file to the PARM disk (CF1) and update the SYSTEM CONFIG file if you have the IODF file specified. Keep the old PRODIODF file on the disk as well. Thus, if the system is IPLed, it will pick up the latest production IODF.

6. Generate an IOCDS from the production IODF using the REXX utility **CBDSIOCP**. This will write out the IOCDS for next IML.

Chapter 6. How to exchange IODF or IOCP files between z/VM and z/OS

In the scenarios previously described, you need to transfer files (data sets) between z/OS and z/VM. To achieve this task, you can select from the following alternatives:

- To transfer an IODF from z/OS to z/VM, you can use the z/OS HCD function *Export IODF*:

  1. From the Primary Task Selection Panel, select *6. Maintain I/O definition files*.

  2. Then select *5. Export I/O definition file* from the **Maintain I/O Definition Files** task menu. This invokes the *Export IODF* dialog where you specify the target z/VM node and user ID. HCD sends the IODF directly to z/VM where you receive the file on a CMS minidisk in a format known to z/VM and observing the IODF file naming conventions:

     filename: IODFxx
     filetype: xxxxIODF

- You can transfer the IODF from z/OS to z/VM via FTP. First use the *Export IODF* dialog as described above and specify an asterisk (*) for both the target user ID and node. This creates a sequential file on z/OS from the z/OS VSAM IODF data set using the following naming conventions:

  `tsoid.EXPORTED.iodfname`

  where:

  tsoid
      is the sending TSO user ID or the TSO prefix, (or is determined by the profile option EXPORTED_HLQ)

  iodfname
      is the part of the IODF data set name after the high-level qualifier.

  Example:

  `DOCU.EXPORTED.IODFCC.WORK`

  Now you can send this file to z/VM using FTP.

- To transfer an IODF from z/VM to z/OS, use the `CMS SENDFILE` command or the **CBDSEXP** REXX utility.

If you want to view your z/OS IODF data set from a zVM system, you can share a volume between z/OS and z/VM. Export the z/OS IODF data set into a sequential file and place it on this shared disk. Then you can look at it from z/VM.

Here are the steps for exporting the IODF and installing the shared DASD:

1. Bring up the z/OS guest on z/VM.

2. Invoke HCD and use option 6.5 *Export I/O definition file*.

3. Vary the device offline: `v  xxxx,OFFLINE` (xxxx denotes the device number of the shared DASD).

4. Detach xxxx from z/OS

5. Then on z/VM:
   ATTACH xxxx * R/O

or link to the appropriate full-pack minidisk to get access to the shared volume that contains the z/OS sequential dataset.

6. ACCESS xxxx T
7. LISTDS T shows the files
8. FILEDEF IODFIN T DSN userid.exported.iodf01 (DSORG PS
9. FILEDEF IODFOUT DISK IODF01 IODF A
10. MOVEFILE IODFIN IODFOUT

Note: The CMS FILE IODF01 IODF A should be re-named to IODF01 WORKIODF A or IODF01 PRODIODF A for use by z/VM HCD.