# Using HCD/HCM to manage your I/O configuration in z/VM

Bertold Reddemann (IBM), Friedrich Beichter (IBM)
03/03/2010

# Table of Contents

# Trademarks


**The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.**

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

**The following are trademarks or registered trademarks of other companies.**

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

**Notes**:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

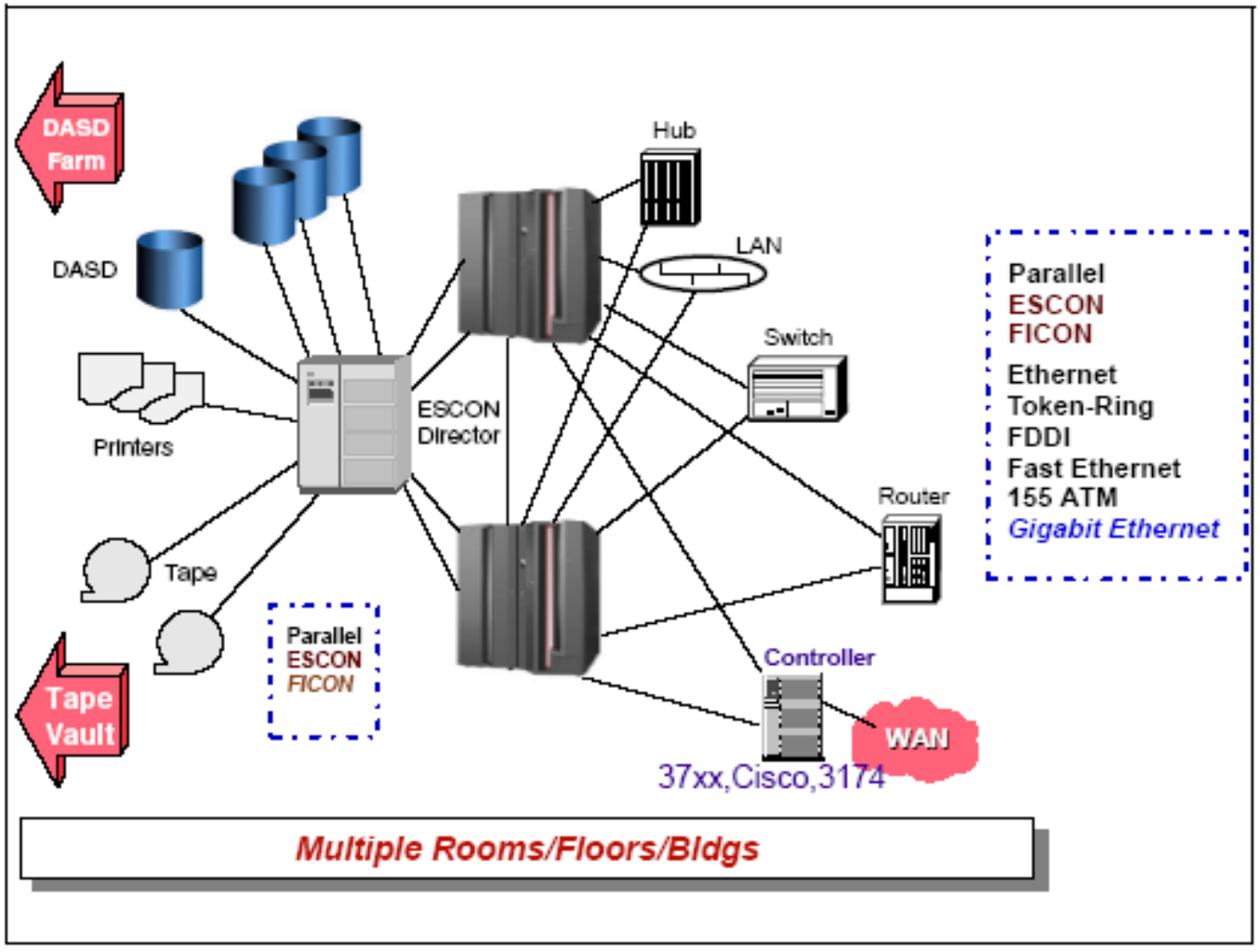
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

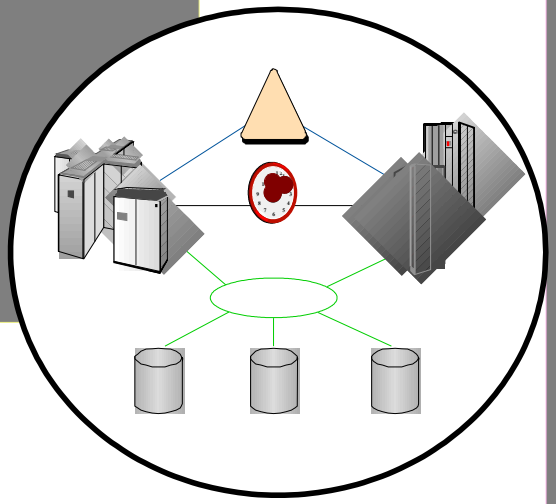
# Scope of I/O Configuration



# I/O Configuration Management

Configuration Management

- Plan
- Define
- Activate
- Operate

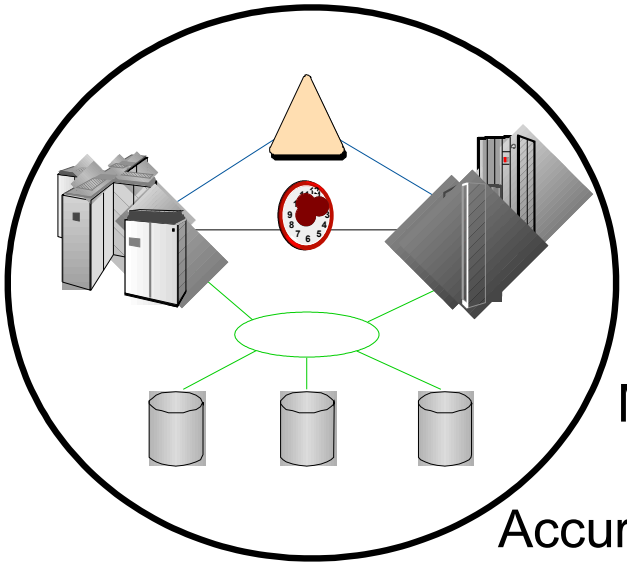


Configuration Aspects

Hardware Management

- assets
- CSS and OS
- physical
- cabling
- network
- floorspace
- power, cooling, etc.
- change
- performance
- problem

# The problem



Different tools

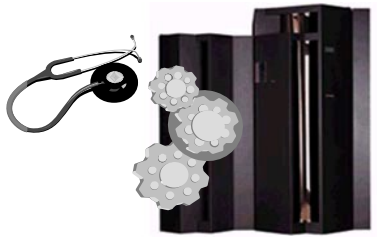
Redundant data

Manual mapping between various aspects

Accuracy of documentation by human discipline only

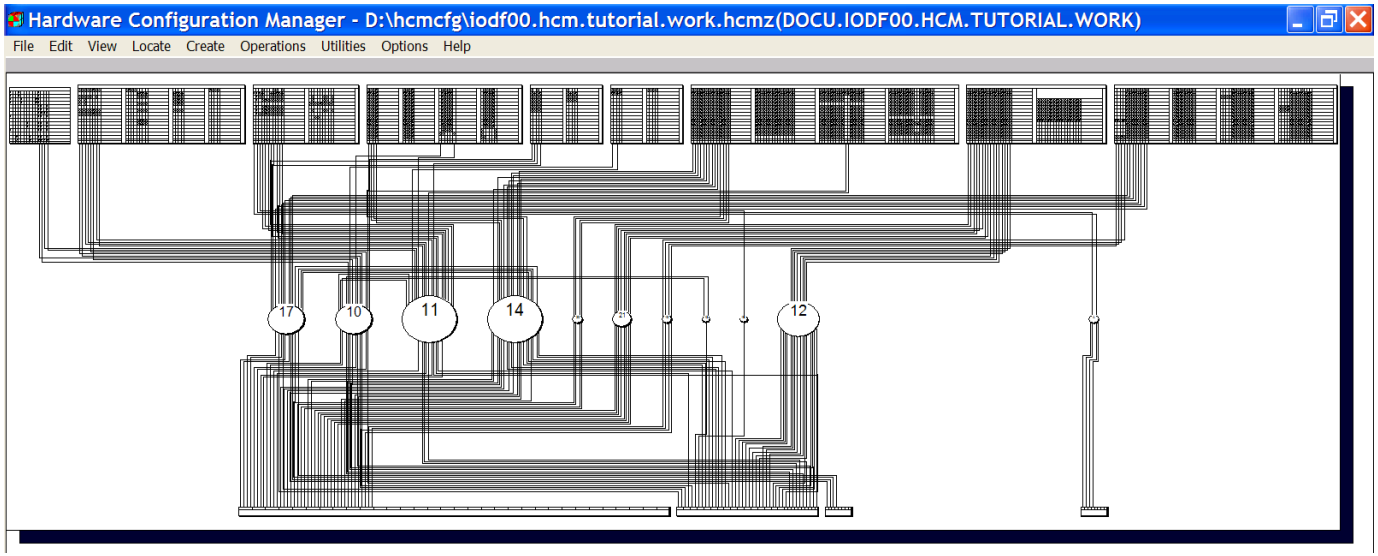
Increasing complexity with System z

# Value of HCD



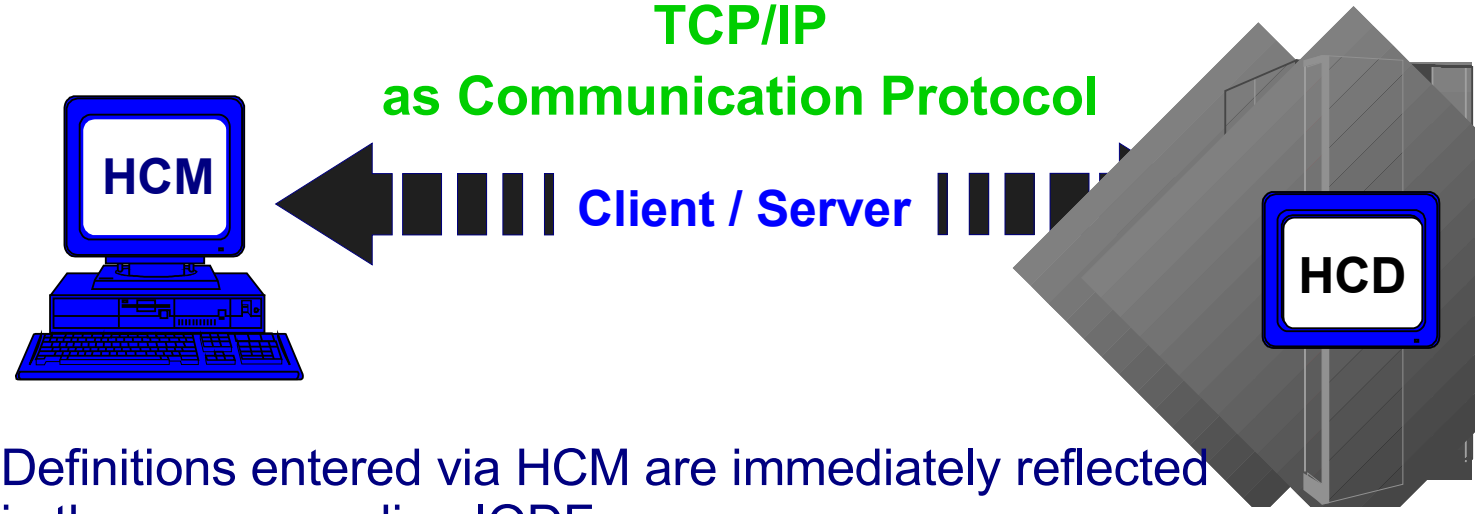
- Integrates **I/O definitions** for the CSS and the OS
- Configuration data stored in a **single repository** (IODF)
- **Rigorous validation** of hardware configuration data
- Helps to **avoid system outages** due to definition inconsistencies
- **Dynamic activate** of configuration changes **non-disruptively**
- Extensive **reporting facility** for configuration data
- HCD Customers (base element in both z/OS and z/VM)
  - **all z/OS customers** (mandatory)
  - **all z/VM customers** (optional)
- **Provides configuration migration**
  - z10 migration
  - configuration migration of IOCP .

# Value of HCM



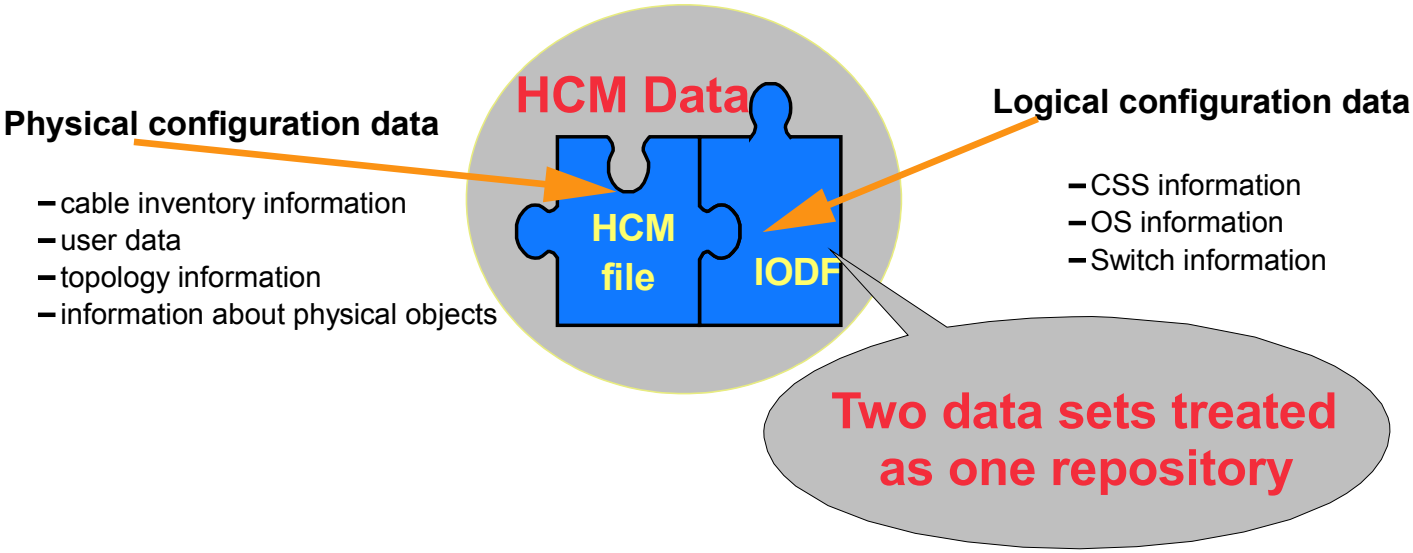

- provides intuitive graphical user interface to HCD
- manages physical and logical data in one place
- includes powerful definition wizards
- provides accurate configuration documentation
- uses HCD as a server for data validation

# HCD – HCM relationship




- ★ Definitions entered via HCM are immediately reflected in the corresponding IODF.
- ★ The access to the IODF as well as the validation of data is done by HCD.
- ★ The use of HCD's rigorous validation algorithm ensures that all data are correct and complete.

# Data repository

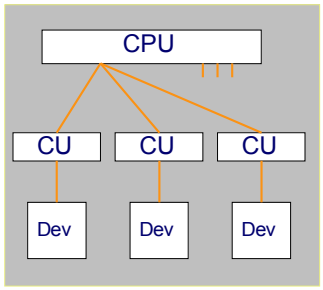


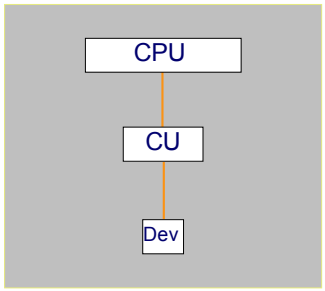

No duplication of data entry for physical and logical definitions

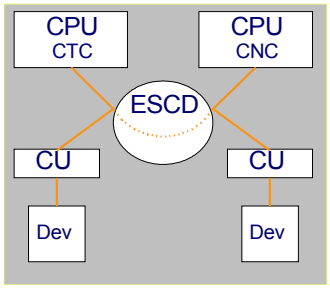
Multiple configuration versions supported
- name of IODF stored in HCM file

Data integrity ensured by time stamp / token
- token updated by HCD whenever IODF is modified
- in case of token mismatch a resynchronization takes place

# General concept

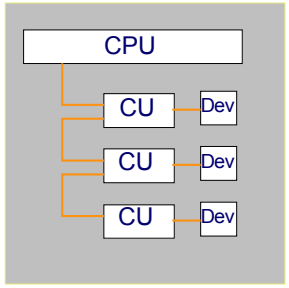


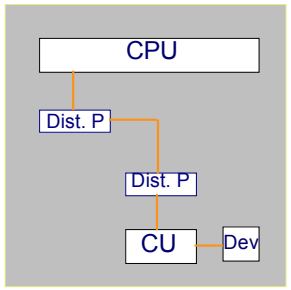



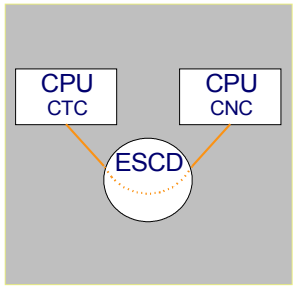




Logical vs. Physical

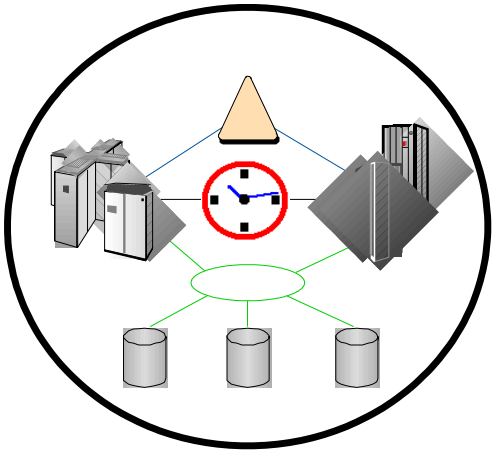
View



# Supported I/O equipment



## Support for I/O Equipment is determined by:

- HCD's Unit Information Modules **(UIMs)**

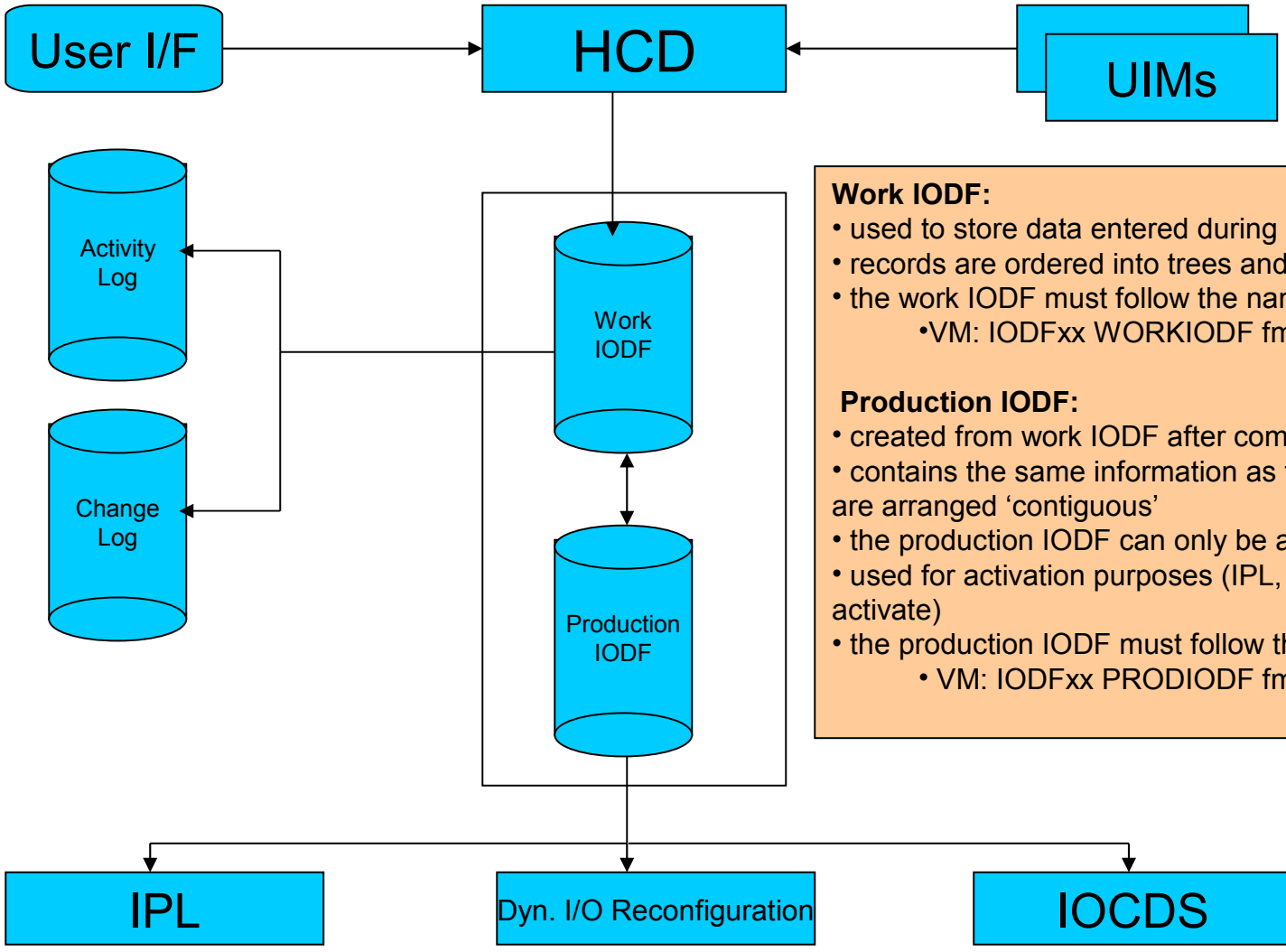
  the logical characteristics and connectivity rules for each device or device group

- HCM's Physical Description Files **(PDFs)**

  the physical appearance of the object on the diagram (e.g. number and name of channel interfaces)

**Same technique for OEM equipment**

# Concept of work and production IODF




**Work IODF:**
- used to store data entered during definition
- records are ordered into trees and connected together
- the work IODF must follow the naming conventions:
  - VM: IODFxx WORKIODF fm

**Production IODF:**
- created from work IODF after completeness validation
- contains the same information as the work IODF but the records are arranged 'contiguous'
- the production IODF can only be accessed in READ mode
- used for activation purposes (IPL, IOCDS download, dynamic activate)
- the production IODF must follow the naming convention:
  - VM: IODFxx PRODIODF fm

# HCD Functions

Controlled by user configuration file, CBDCONF NAMES

- Create and initialize an empty IODF
  - `cbdsiodf iodf01 workiodf a 2096`
- Migrate (import) I/O configuration statements / IOCP deck into an IODF
  - `cbdsmigr iodf01 workiodf a iocp deck a i proc1 2097-e12 lpar`
- Export I/O configuration statements from an IODF
  - `cbdsconf iodf01 workiodf a config deck a i proc1`
- Extract software configuration file from RDEV control blocks
  - `cbdsrdev cbdmconf rdevlist a`
- Import RDEV configuration into the IODF
  - `cbdsmigr iodf01 workiodf a cbdmconf rdevlist a r vm1`
- Build a production IODF
  - `cbdsprod iodf01 workiodf a iodf01 prodiodf a`
- Build a work IODF from a production IODF
  - `cbdswork iodf01 prodiodf a iodf01 workiodf a`

# HCD Functions (cont.)

- Copy IODF
  - `cbdscopy iodf01 workiodf a iodf11 workiodf c (repl`
- Print configuration reports
  - `cbdsrep iodf01 workiodf a iodf01 report a cpuds`
- Create a graphical configuration report
  - `cbdsdraw iodf01 workiodf a iodf01 script a lcu proc1`
- Compare configurations in IODFs
  - `cbdscomp iodf01 prodiodf a iodf11 prodiodf c iodf01 compare a`
- Export an IODF
  - `cbdsexp iodf01 prodiodf a jack boevm3`
- Import an IODF
  - `cbdsimp 0018 iodf88 workiodf a`

# HCD Functions (cont.)

- Build an IOCP deck
  - `cbdsiocp iodf01 prodiodf a proc1 d proc1 iocp a`
- Write an IOCDS
  - `cbdsiocp iodf01 prodiodf a proc1 i a1`
- Dynamically activate a configuration
  - `cbdsact iodf01 proc1 vm1 a1`

- Start the z/VM HCD TCP/IP dispatcher for HCM
  - `cbdsdisp`
- Stop the z/VM HCD TCP/IP dispatcher for HCM
  - `server stop`

# Additional CMS files

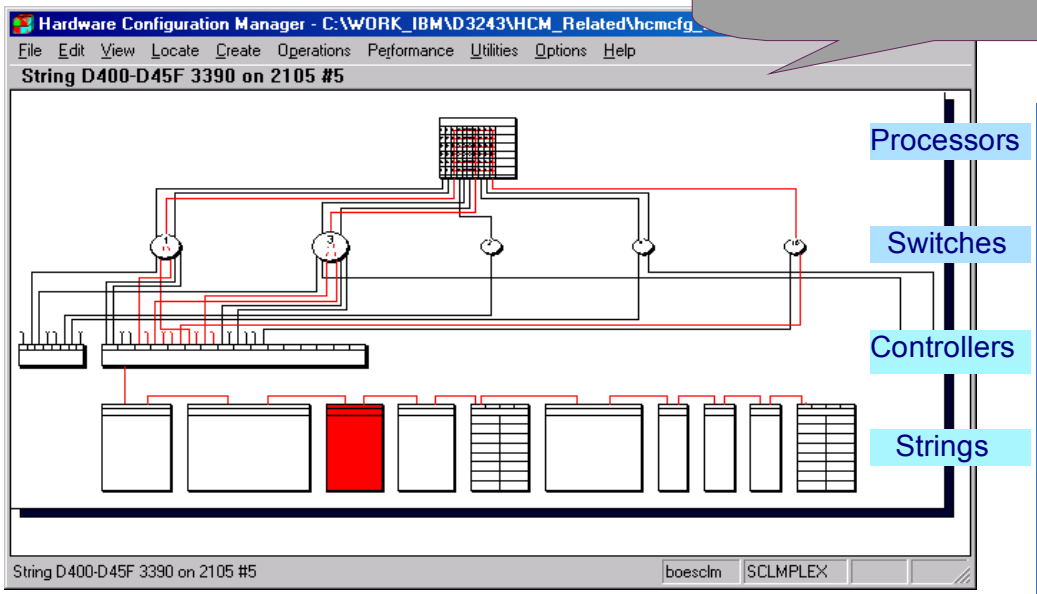
- HCD message log `<rexx-utlility_name> msglog a`
- HCD profile `hcd profile a`
- HCD trace file `hcd trace a`
- HCD activity log file `<iodf_name> workactl a`
- HCD change log file `<iodf_name> workclog a`
- HCM master configuration file

  `<iodf_name> workmcf a`

  `<iodf_name> prodmcf a`

# Layout of the HCM screen

The logical definitions in the IODF are represented as
- **connections between objects in the diagram** , and
- **attribute information in appropriate dialogs**

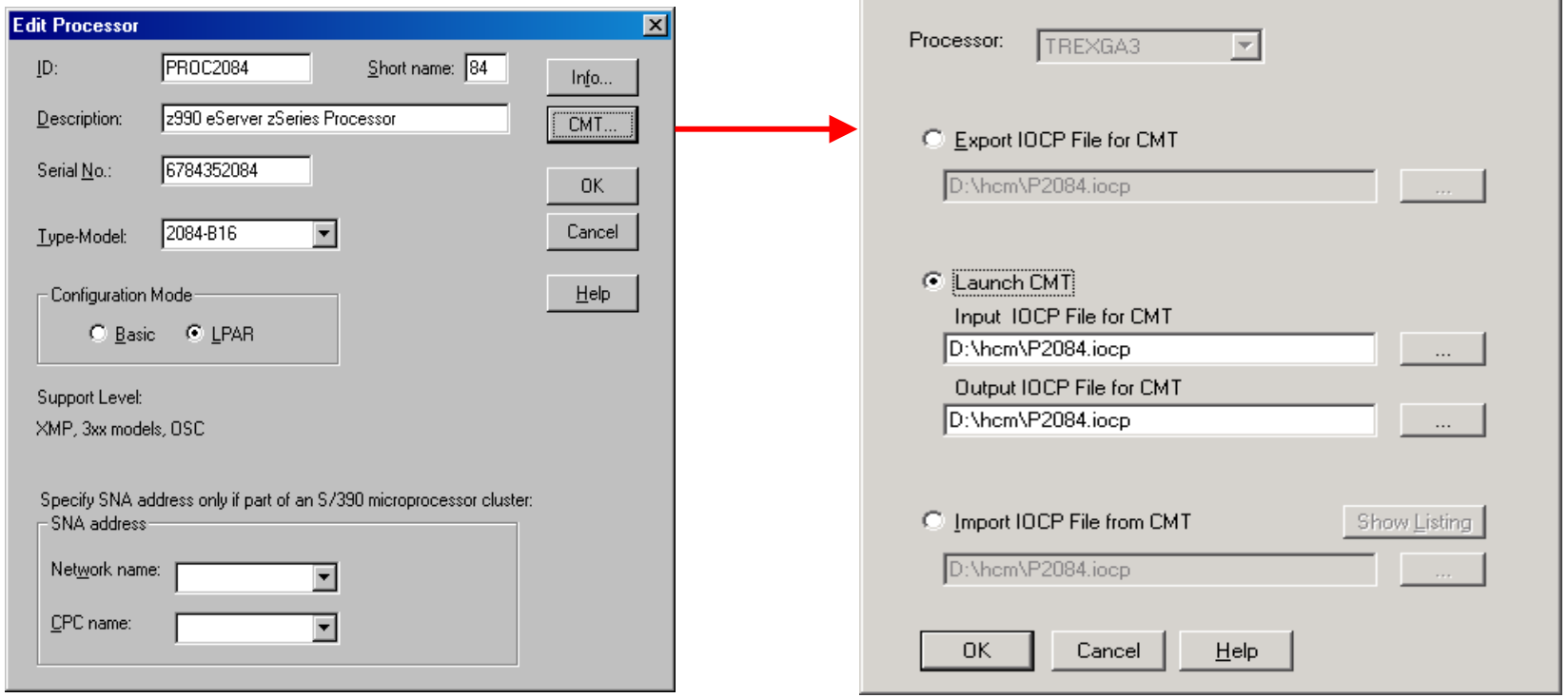



Use HCM to add, delete, or modify physical connections and associated logical definitions.

## Configuration Diagram

- **starting point** for any configuration task (like create/edit, or connect/disconnect objects)
- presents the **physical view** of the elements and their relation
- layouted in a **hierarchical structured** form (Processors, Switches, Controllers (CU), and Strings)
- allows **zooming and filtering** techniques to work with the elements of interest

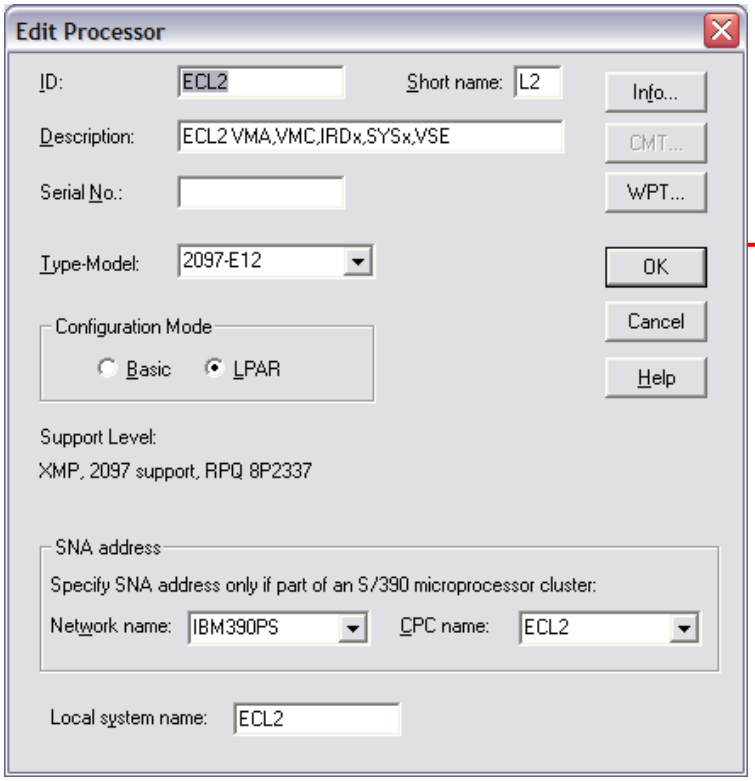
# Support of CHPID Mapping Tool

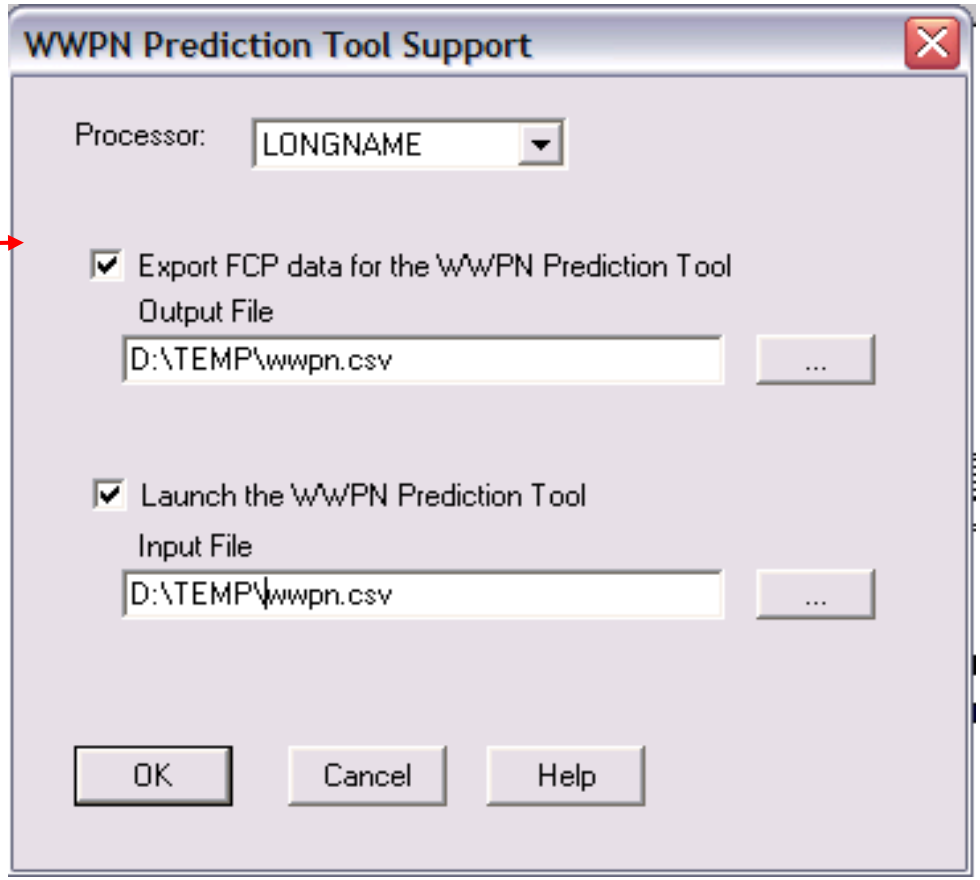



CHPID Mapping Tool (CMT): Assigns PCHIDs to CHPIDs 'automatically'
'CHPID Mapping Tool Support' dialog in HCM assists you in

1. Exporting an IOCP file for the CMT for a selected processor,
2. Launching the CMT with IOCP input and output specfications
3. Importing an updated IOCP file from the CMT to the IODF.

# Support of WWPN Prediction Tool




HCD offers a new option for exporting the FCP specific part of the I/O configuration. This file can be used together with the I/O Serial Number (from Resource Link) as input for the WWPN Prediction Tool to generate the WWPNs for the system.

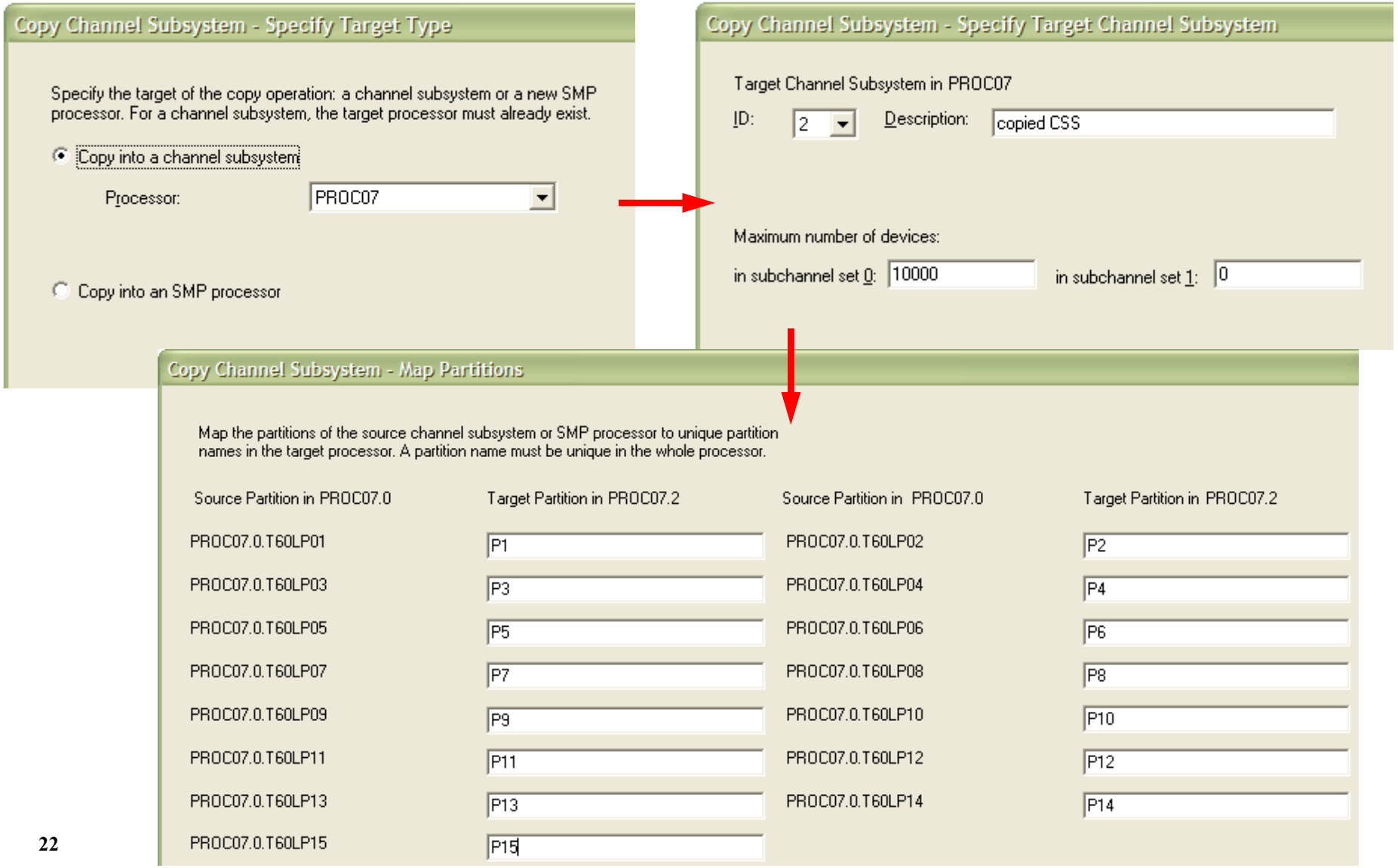
# HCM Wizards

To support you in performing complex processor configuration tasks, HCM offers several wizards or utilities:

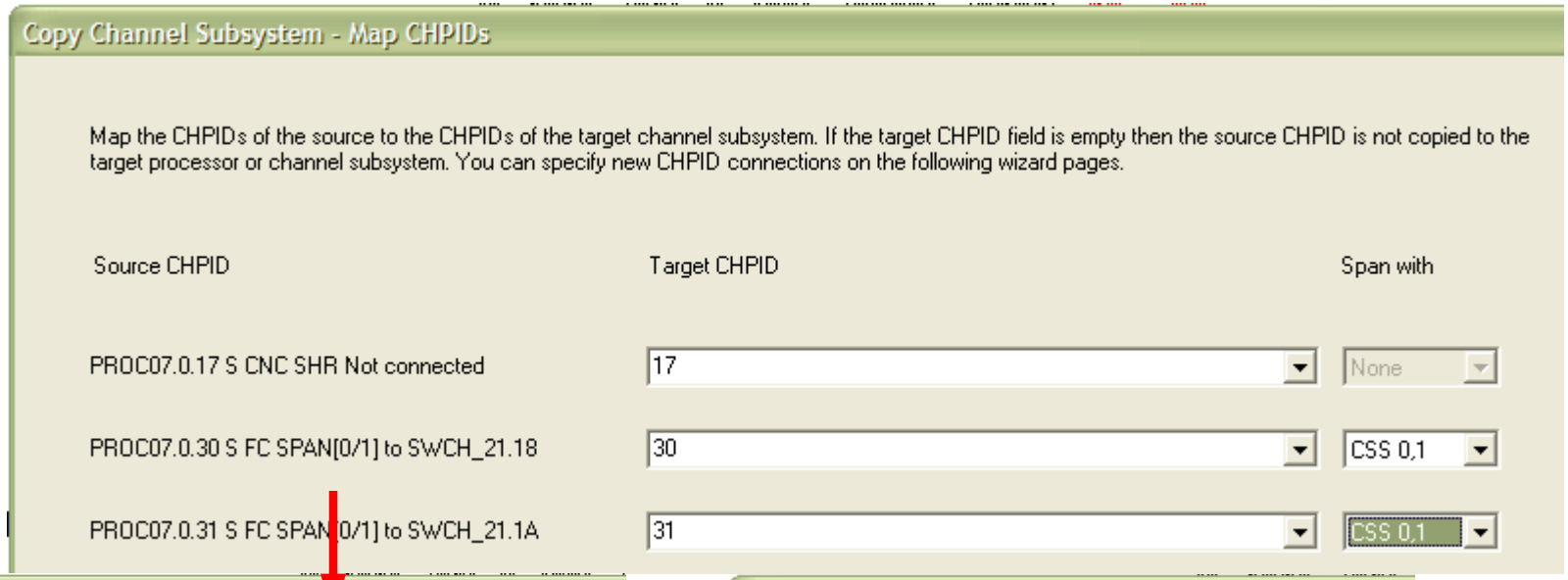
- Copy Processor
- Copy Channel Subsystem (*)
- Copy Partition
- Aggregate CHPID
- Move Port Connections
- Import/Export PPRC Connections
- Converting CNC Channels to FICON Channels
- Change Link Address Format
- Creating/Copying/Editing an I/O subsystem (*)
- Define CTC / CF / STP connections

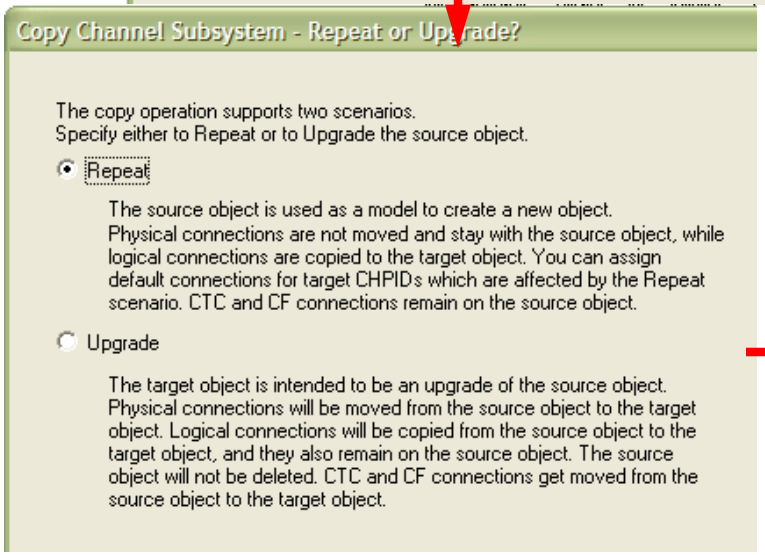
(*) Example dialogs are shown on the following pages

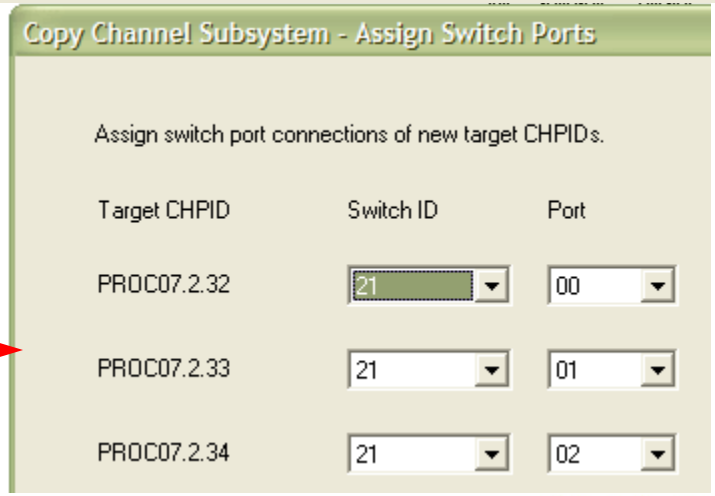
# Copy channel subsystem (1)



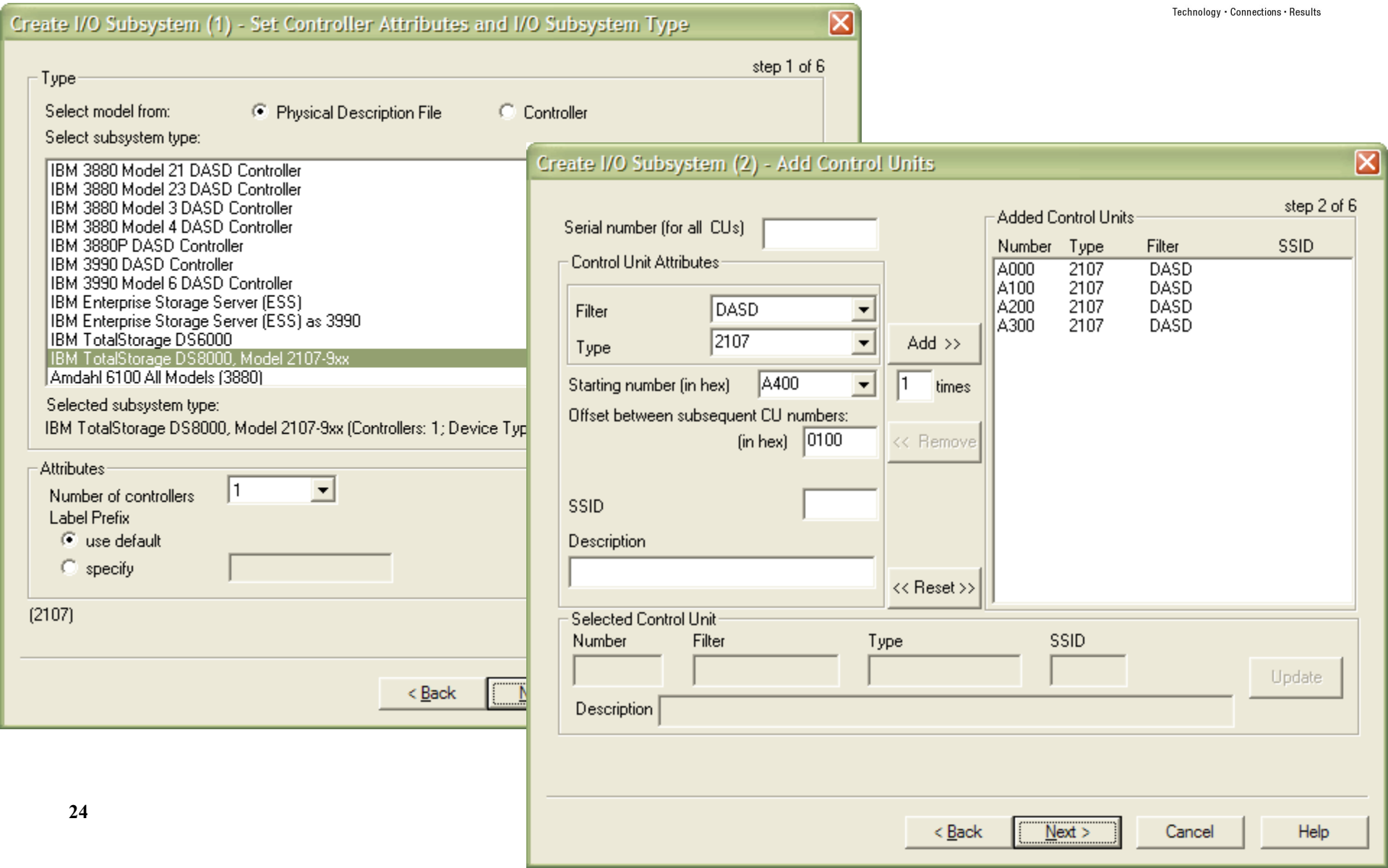
# Copy channel subsystem (2)



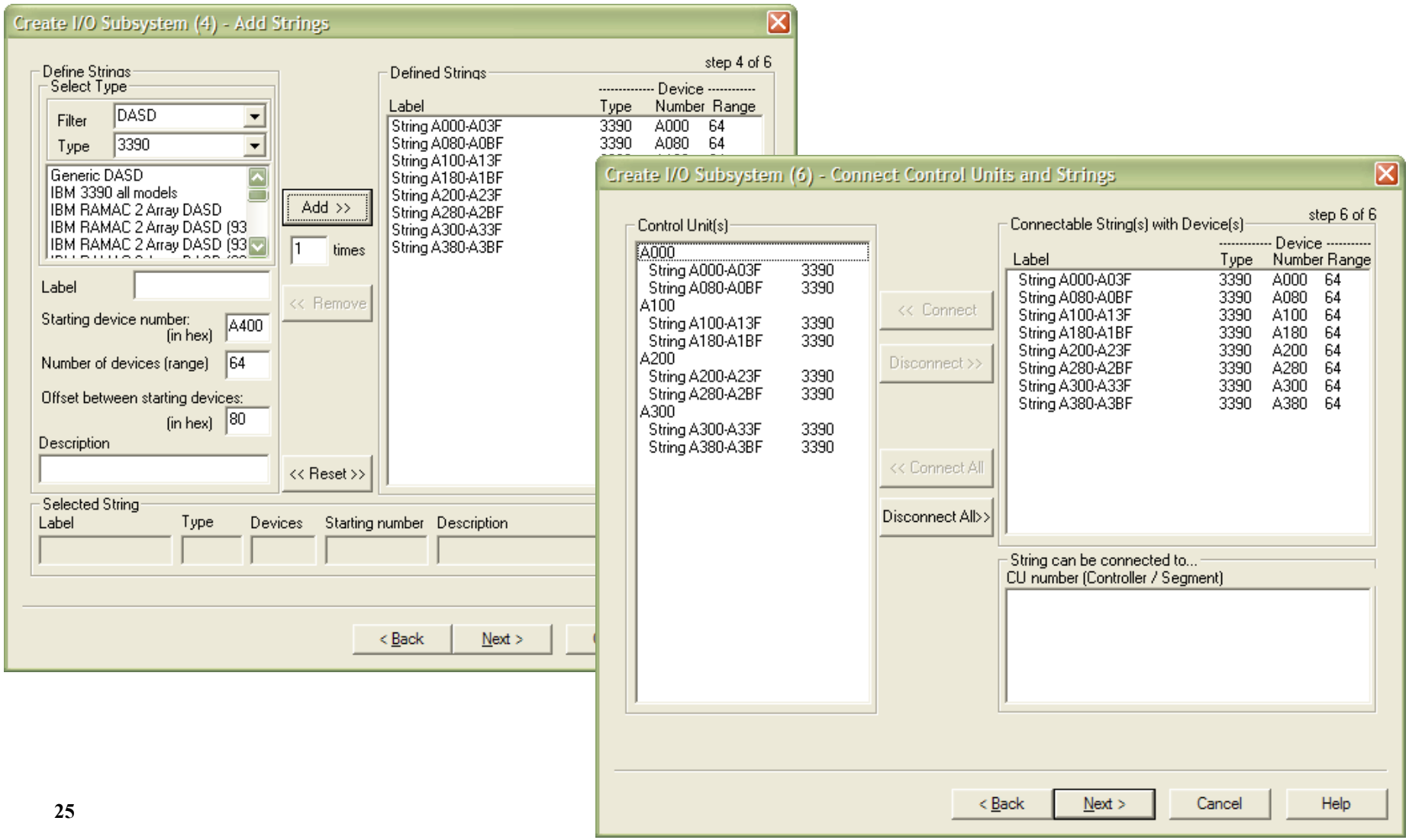



# Create I/O subsystem (1)



# Create I/O subsystem (2)



# Reports

- HCD Reports
  - Textual Reports: CSS configuration (like IOCP reports), OS configuration, switch configuration, CTC connections, CF connections, supported hardware, compare reports
  - Graphical Reports (logical view): bookmaster, DCF, GML, GDF
- HCM Reports
  - HCD textual reports
  - Textual: processor, switches, controllers, strings, cables and links, cabinets and general boxes, crossbar switches
  - Format: print, XML, csv
  - Graphical reports (physical view): diagrams

# Migration to z/VM HCD

- Change the dynamic I/O configuration method from CP commands to lock-step mechanism of HCD.

- Two scenarios:

1. Installation has no z/OS system:
   > IODF definition for z/VM has to be done by z/VM HCD

2. Installation has a z/OS system
   > IODF definition for z/VM can be done by z/OS HCD

# Scenario 1: z/VM is used to define the IODF

- Step 1: Prepare your IOCP deck for migration to the IODF
- Step 2: Migrate the IOCP deck
- Step 3 (optional): Migrate your RDEV information
- Step 4: Verify the IODF configuration
- Step 5: Prepare for POR and IPL with the production IODF
- Step 6: POR with HCD generated IOCDS
- Step 7: IPL with IODF
- Step 8: Load IODF into HCM

- Recurring step: Perform dynamic I/O reconfiguration

# Step 1: Prepare your IOCP for migration

- Use comment tags ***$HCDC$** and ***$HCD$**
- RESOURCE statement:
  - Include partition usage types

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

- CHPID statements:
  - Provide switch connection data
  - Provide coupling facility connection data

```
   CHPID  PATH-(CSS(0),14),SHARED,PCHID=100,TYPE=FC
*$HCDC$                   SWPORT=(A0,18)
```

- CNTLUNIT statements:
  - Change UNIT operand to control unit type that is supported by HCD
  - Provide switch connection data
  - Provide serial numbers for DASD/TAPE control units

```
   CNTLUNIT CUNUMBR=8000,PATH=(14,16,19,1B),LINK=(E0,E4,E8,EC),     *
            CUADD=0,UNITADD=(00,64)),UNIT=2107
*$HCDC$                   SWPORT=((A0,E0),(A0,E8),(B0,E4),(B0,EC)),SERIAL='23425'
```

- IODEVICE statements:
  - Change UNIT/MODEL operands to device types that are supported by HCD
- Include SWITCH statements for the used directors

```
   *$HCD$   SWITCH  SWID=A0,ADDRESS=60,PORT=((04,43),(FE,FE)),UNIT=2032
```

# Step 2: Migrate the IOCP deck to IODF

- Create and initialize an empty work IODF
  - `cbdsiodf iodf71 workiodf h 4096`
- Migrate the prepared work IOCP deck into the work IODF
  - `cbdsmigr iodf71 workiodf h proc1 iocp a proc1 2097-e12`
- Verify successful migration
  - Inspect the output files:

| | |
|---|---|
| CBDSMIGR MSGLOG | HCD message log file informs about the success of the migration function |
| CBDSMIGR LISTING | Assembler listing file informs about syntax errors |
| CBDSMIGR MESSAGES | Migration messages, inform about validation errors |

- Repeat the migrate task until the IOCP has been successfully imported in the IODF
- Produce an HCD report and check the configuration for correct data
  - `cbdsrep iodf71 workiodf h iodf71 report a`

# Step 3 (optional): Migrate RDEV data

- Create an RDEVLIST from your active z/VM system
  - `cbdsrdev vmconf rdevlist a`
- Migrate the RDEVLIST into your work IODF
  - `cbdsmigr iodf71 workiodf h vmconf rdevlist a r realvm`
- Complete the OS configuration
  - Export OS configuration as I/O configuration statements

    `cbdsconf iodf71 workiodf h realvm deck a o realvm`

  - Update OS configuration with additional devices

    `RDEVICE DEVNO=(1100,1),DEVTYPE=3215,OFFLINE=NO`

  - Re-migrate the updated OS configuration deck

    `cbdsmigr iodf71 workiodf h realvm deck a op realvm`
- Verify the OS configuration
  - Generate an OS configuration report

    `cbdsrep iodf71 workiodf h realvm report a mn`

# Step 4: Verify I/O configuration

- Check whether CSS definitions match the OS definitions
  - Use the HCD compare report to check partition configuration with OS configuration

```
cbdscomp iodf71 workiodf h * * * iodf71 compare a d
                                    proc1 lp1 realvm
```

- Build the production IODF
  - `cbdsprod iodf71 workiodf h iodf71 prodiodf h`
- Inspect the messages given in the HCD message log file
  - `CBDSPROD MSGLOG A`
  - Error messages does not allow the production IODF to be built
  - Warning messages/information messages may indicate configuration problems
- Correct the IODF definitions if necessary

# Step 5: Prepare for POR and IPL with production IODF

- Perform an IOCDS download
  - `cbdsiocp iodf71 prodiodf h proc1 i a1`
- Copy the production IODF to the SYSPARM disk
- Set up the SYSTEM CONFIG file
  - Insert IODF statement

    `IODF IODF71 REALVM`

    or

    `IODF * REALVM`

# Step 6: POR with HCD generated IOCDS

- Adapt the Reset Profile to allow dynamic changes.
- Select the IOCDS slot to which the IODF processor configuration has been written.
- Perform a POR.

# Step 7: Perform IPL

- IPL your system
- After IPL, check if HCD controls the I/O configuration
  - `q hcd`

```
HCD is currently active: IODF = IODF71 PRODIODF
HCD is enabled for dynamic hardware changes
HCD is controlling the software configuration
HCD recover is not currently required
Ready;
```

  - `q token`

```
PROC1…m.o……10-03-1007:50:15SYS4    IODF71
Ready;
```

- If IODF to be loaded is not on the SYSPARM disk, a disabled WAIT occurs.

# Step 8: Load IODF into HCM

- Download HCM from host and run installation EXE file.
- On z/VM host, start HCD dispatcher
  - `cbdsdisp`
- Launch HCM from your workstation
- Logon to the z/VM user, using TCP/IP host name and port number
- In **`File`** menu, select item **`IODFs`**…
- On the **`IODFs`** list, select the work IODF and press the LOAD button
- Provide the name of the HCM configuration file
- You now see the default configuration diagram of the IODF and are now ready to work with HCM.
- For new logins, select **`File – Open`** to open the HCM configuration file

# Recurring step: Perform dynamic I/O


- Using HCD, build a new production IODF with the changes:
  - Make a work IODF from your active production IODF
    ```
    cbdswork iodf71 prodiodf h iodf72 workiodf h
    ```
  - Export processor and OS configurations to CMS files
    ```
    cbdsconf iodf72 workiodf h proc1 deck a i proc1 (repl
    cbdsconf iodf72 workiodf h realvm deck a o realvm (repl
    ```
  - Edit the I/O configuration statements with the changes
  - Remigrate the changed configuration files (partial migrate)
    ```
    cbdsmigr iodf72 workiodf h proc1 deck a ip proc1
    Cbdsmigr iodf72 workiodf h realvm deck a op realvm
    ```
  - Build production IODF
    ```
    cbdsprod iodf72 workiodf h iodf72 prodiodf h
    ```
- Or, use HCM to perform the changes to the IODF directly

- Activate the changes
  - `cbdsact iodf72 proc1 realvm a1`
- Use `q hcd` and `q token` to check that the new configuration is active.

# Scenario 2: z/OS is used to define the IODF

Almost same steps as with scenarios 2, however, they may be done differently.

- Step 1: Prepare your z/VM IOCP deck for migration to the IODF
- Step 2: Migrate the IOCP deck
- Step 3 (optional): Migrate your RDEV information
- Step 4: Verify the IODF configuration
- Step 5: Prepare for POR and IPL with the production IODF
- Step 6: POR with HCD generated IOCDS
- Step 7: IPL with IODF
- Step 8: Load IODF into HCM

- Recurring step: Perform dynamic I/O reconfiguration

## Step 1: Prepare your IOCP for migration

- identical to scenario 1

## Step 2: Migrate the IOCP deck to the IODF

- Transmit the IOCP deck from your z/VM to your z/OS system
  - `Sendfile proc1 iocp to ossys at boetrx2`
- Migrate the VM IOCP deck using z/OS HCD into the existing work IODF

# Step 3 (optional): Migrate your RDEV data


- Export your work IODF to z/VM using z/OS HCD
- Import the work IODF under your z/VM system
  - `cbdsimp 0018 iodf71 workiodf h`
- Create an RDEVLIST from your active z/VM system
  - `cbdsrdev vmconf rdevlist a`
- Migrate the RDEVLIST into your work IODF
  - `cbdsmigr iodf71 workiodf h vmconf rdevlist a r realvm`
- Complete the OS configuration
  - Export OS configuration as I/O configuration statements

    `cbdsconf iodf71 workiodf h realvm deck a o realvm`
  - Update OS configuration with additional devices

    `RDEVICE DEVNO=(1100,1),DEVTYPE=3215,OFFLINE=NO`
  - Re-migrate the updated OS configuration deck

    `cbdsmigr iodf71 workiodf h realvm deck a op realvm`
- Verify the OS configuration
  - Generate an OS configuration report

    `cbdsrep iodf71 workiodf h realvm report a mn`
- Export the work IODF back to z/OS
  - `cbdsexp iodf71 workiodf h ossys boetrx2`

# Step 4: Verify I/O configuration

- If you are using HCM with z/OS HCD, open your HCM configuration file connected to your work IODF. A resync will take place.
- Other tasks are same as with scenario 1 but tasks are done in z/OS HCD or z/OS HCM.

# Step 5: Prepare for POR and IPL with production IODF

- Perform remote IOCDS download under z/OS HCD for the CEC running your z/VM system
- Export the production IODF to your z/VM system and receive it there
- Copy the production IODF to the SYSPARM disk
- Set up the SYSTEM CONFIG file as with scenario 1

## Step 6: POR with HCD generated IOCDS

- Same as with scenario 1

## Step 7: Perform IPL

- Same as with scenario 1

## Step 8: Load the IODF into HCM

- If you are using HCM with z/OS HCD, open your HCM configuration file connected to your work IODF. A resync will take place to include the new processor configuration into the HCM configuration and diagram.

# Recurring step: Perform dynamic I/O

- Using the z/OS HCD or z/OS HCM dialogs, build a new production IODF with the changes for your z/VM CPC or VM operating system.
- Distribute the production IODF to your z/VM CPC.
- Activate the changes
  - `cbdsact iodf72 proc1 realvm a1`
- Use `q hcd` and `q token` to check that the new configuration is active.

# Exchange an IODF between z/VM and z/OS

- Export an IODF from z/OS to z/VM
  - z/OS HCD function ***Export IODF*** (dialog option 6.5). Use **`CMS RECEIVE`** or the ***CBDSIMP*** REXX utility to obtain the CMS file.
  - Use FTP to the exported VSAM data format
  - Use z/OS HCM function ***Export IODF*** to download the exported format to the HCM workstation. Send the file as mail attachment or connect HCM to z/VM. Use HCM function ***Import IODF*** to establish the IODF as CMS file.
- Export an IODF from z/VM to z/OS
  - Use **`CMS SENDFILE`** or the ***CBDSEXP*** REXX utility. On z/OS, use HCD dialog option 6.6. (***Import IODF***).
  - Use FTP to the CMS file. On z/OS, use HCD dialog option 6.6.
  - Use HCM function ***Export IODF*** to download the IODF to the workstation. Connect to z/OS HCD. Use HCM function ***Import IODF*** to establish the IODF as VSAM file.
- Sharing volume between z/OS and z/VM

1. Bring up z/OS guest on z/VM.
2. Invoke z/OS HCD and use option 6.5 ***Export IODF*** to shared DASD xxxx.

```
1.VARY xxxx,OFFLINE
```

1. Detach device xxxx from z/OS.

On z/VM:

```
5.ATTACH xxxx * R/O or link to the appropriate full-disk minidisk
6.ACCESS xxxx T
7.LISTDS T
8.FILEDEF IODFIN T DSN userid.exported.iodf01 (DSORG PS
9.FILEDEF IODFOUT DISK IODF01 PRODIODF H
10.MOVEFILE IODFIN IODFOUT
```

# Session Summary

- This session showed
  - What benefits the use of HCD and HCM for I/O configuration management has.
  - How HCD and HCM can be used to manage the I/O configuration for z/VM.
  - Which are the migration steps to use HCD and HCM
- Q & A

# Appendix

**References**:

- z/VM I/O Configuration, SC24-6100
- z/OS and z/VM Hardware Configuration Manager User's Guide, SC33-7989
- z/OS and z/VM Hardware Configuration Definition Messages, SC33-7986

**HCD/HCM Homepage:**

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