I/O Auto-configuration with IBM zEnterprise using HCD

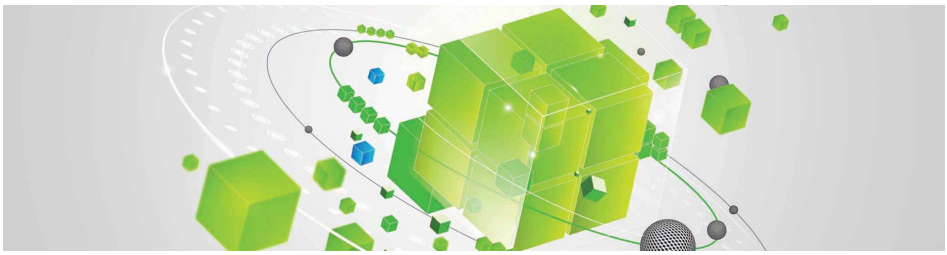
z/OS Discovery And Autoconfiguration (zDAC)

Session 9683

SHARE in Orlando, 2011

August 10th, 2011

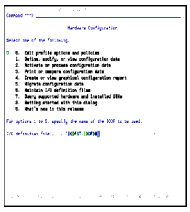
Scott Compton
IBM Corporation
compton@us.ibm.com

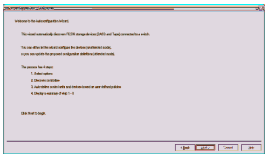


Agenda

- What is Discovery and Autoconfiguration (zDAC)?
- Discovery and Autoconfiguration Overview
- Discovery and Autoconfiguration Techniques
- Tips
- Demo
- Q&A

What is Discovery and Autoconfiguration (zDAC)?



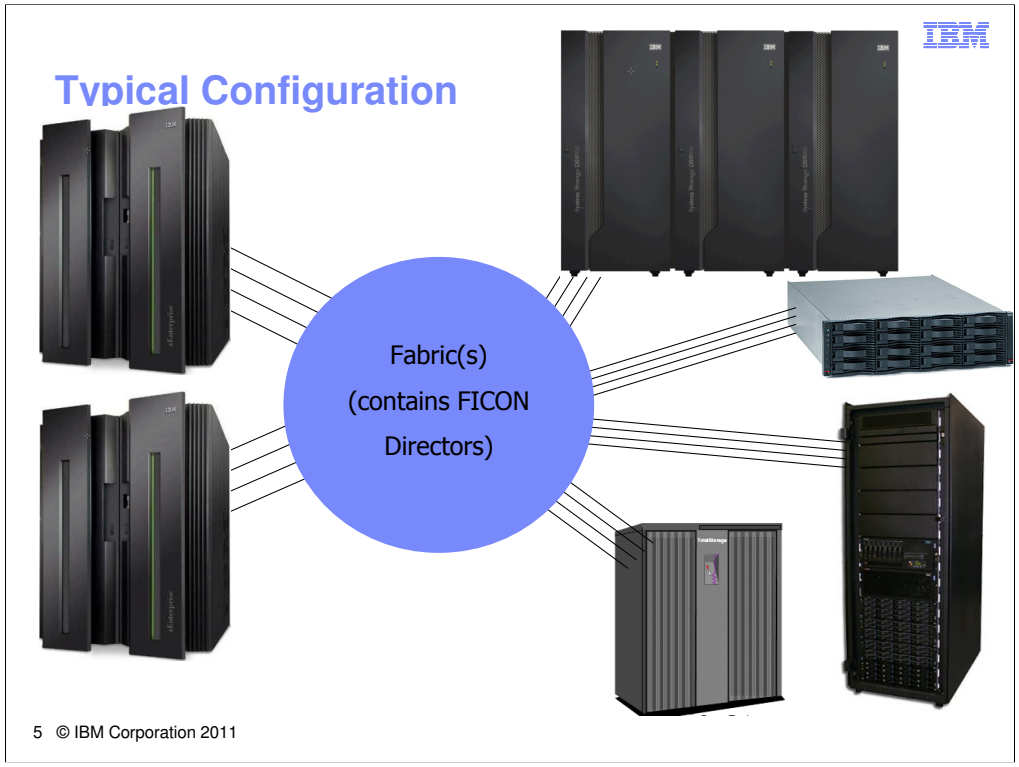


- **New feature of z196**
  - **Invoked as a new option Hardware Configuration Dialog (HCD) and Hardware Configuration Manager (HCM)**
- **Provides capability to discovery attached disk and tape controllers connected in switched FICON fabrics**
- **Detects new controllers (storage subsystems)**
- **Detects new control units on existing controllers**
- **Detects new devices on existing control units**
- **Proposes control unit and device numbering**
- **Proposes paths for all discovery systems to newly discovered control units**

In z/OS Release 12, Hardware Configuration Dialog (HCD), an ISPF application, and Hardware Configuration Manager (HCM), a workstation application, allow for the exploration of switched fibre channel channels in order to determine what physical storage controllers are attached and reachable. Disk and tape controllers can now be automatically detected and added to the I/O configuration.

zDAC Goals

- Reduce complexity and skill required to configure devices
- Reduce the time it takes to make I/O configuration changes
- Ensure the defined configuration aligns with reality
- Ensure that high availability expectations are met in the configuration
- Discover new and older controllers
- Verification Tool



Shown here is a sample configuration. Processors and controllers are interconnected using one or more fabrics.

A fabric consists of one or more interconnected directors or switches.

On the left side of the picture, processors are shown connecting to the fabric(s) with multiple fibre optic cables. On the right, A selection of newer and older Disk and Tape Controllers are shown connecting to the fabric with fibre optic cables

Some Things to Consider…

- **Physical planning is still up to you**
- **Logical definitions on the controller are still up to you**
- **What z/OS images should be allowed to use the new devices?**
- **How should new devices be numbered?**
- **How many paths to new control units should be configured?**

Existing controllers with new control unit definitions can be discovered without physical changes, provided that architectural capacity permits connectivity over existing paths.

New controllers require new cables connecting to ports on either new or existing switches.

The System z connectivity must also be considered – is there enough channel capacity and enough architectural capacity?

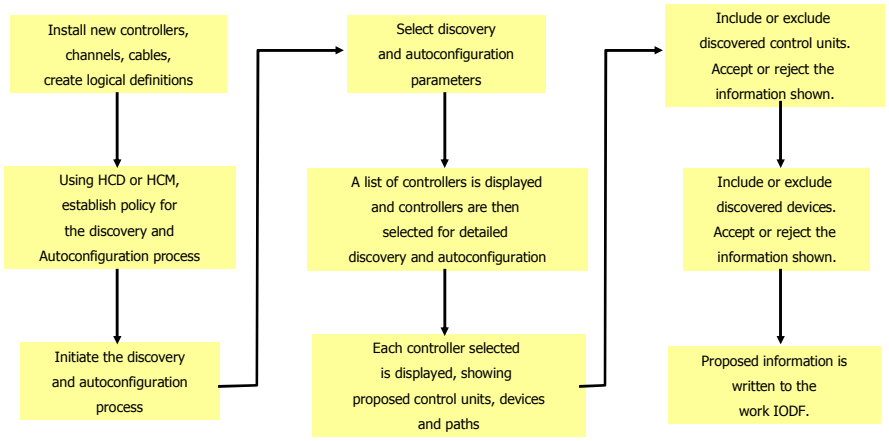
Discovery can be scoped at the sysplex level, or can be a subset of those systems.

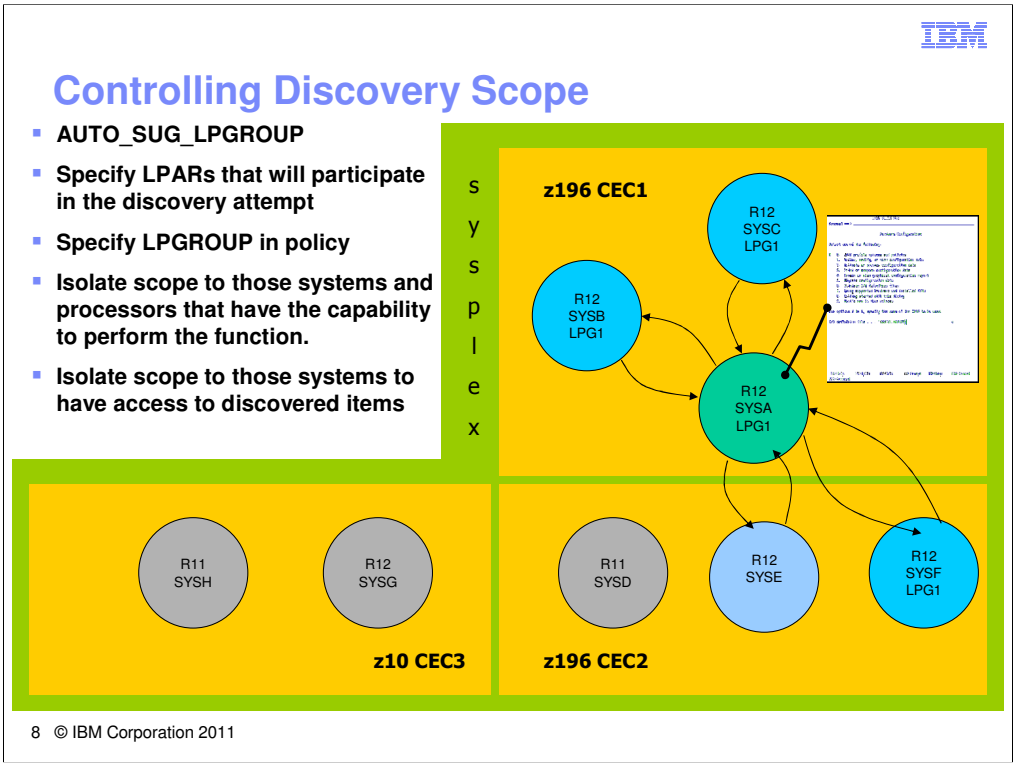
You may also want to control the device number and control unit number ranges that are used.

Finally, you may want to control how many paths you need to ensure that you have the availability and performance characteristics you need for the new devices and control units.

The physical cables and the controller updates are up to you. The rest of the considerations listed here are controlled by you by establishing policy.

Discovery and Autoconfiguration Steps





HCD/HCM policy option AUTO_SUG_LPGROUP allows the user to specify a named LPAR group (LPGROUP) that contains LPARs defined on one or more CECs. It is simply a list of LPARs in a named group. LPGROUPs are created and maintained using HCD and HCM policy options. Many LPGROUPs can be created ahead of time and used as necessary when you wish to control the scope of discovery operations.

In the Autoconfiguration Policy, you may then identify an LPGROUP that will be used during discovery. If no LPGROUP is specified, the default scope is the entire sysplex.

The requirements for an LPAR to have discovery capability are to have:

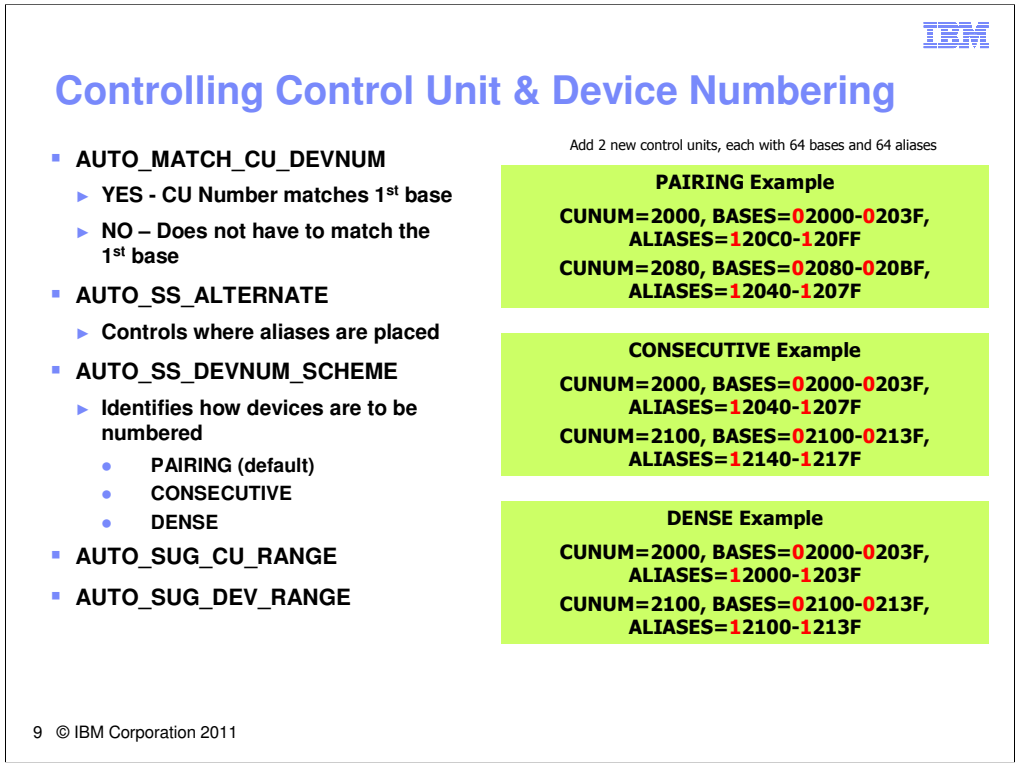
•z/OS Version 1 Release 12

•z196 processor

•At least 1 LPAR at z/OS Version1 Release 12 level with authorization to make dynamic I/O configuration changes.

•User must have authority to make I/O configuration changes (UPDATE access to MVS.ACTIVATE OPERCMDS resource)

The example shown shows LPARs on 3 CECs, two of which are z196 and have discovery capability. On those z196 CECs, some systems are running at z/OS Version1 Release 12. The HCD and HCM capability is present in the z/OS Version 1 Release 12 level. Here, an HCD user running on SYSA can initiate a discovery request using an LPGROUP “LPG1” which contains SYSA, SYSB, SYSC and SYSF. Any proposed control units and devices would only be added to those systems identified in the LPGROUP.



Policy options can help you control the way control units and devices are numbered.

However, if some definitions for the control units are found in the IODF, policy options may not be followed in order to use numbers previously defined in the IODF. For example, you may be discovering on SYSA only, and discovery processing may find that the control units discovered are already defined on SYSB. If possible, those numbers will be used instead of creating new ones using the policy options in effect.

If no existing definitions are found, policy options are used to determine appropriate control unit and device numbers.

AUTO_MATCH_CU_DEVNUM is a policy setting that indicates whether the CU number should match the 1st base UA. The default is YES and is generally the recommended approach.

AUTO_SS_ALTERNATE is a policy option that indicates which subchannel set (0, 1, 2) should be used to contain alias device numbers. In the examples on the right, the option used was “1”

AUTO_SS_DEVNUM_SCHEME controls how alias numbering is to be done. This can also affect how base device numbering is done, depending on how constrained your device number space is. The following options are allowed:

- 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 will be used.
- PAIRING Base and alias device numbers are assigned alternatively starting with for example device numbers xx00 and xx80 versus xx80 and xx00. PAIRING is the default.

Within each numbering scheme, there are many attempts to use each scheme, relaxing scheme rules each time a pass fails to satisfy the current numbering operation.

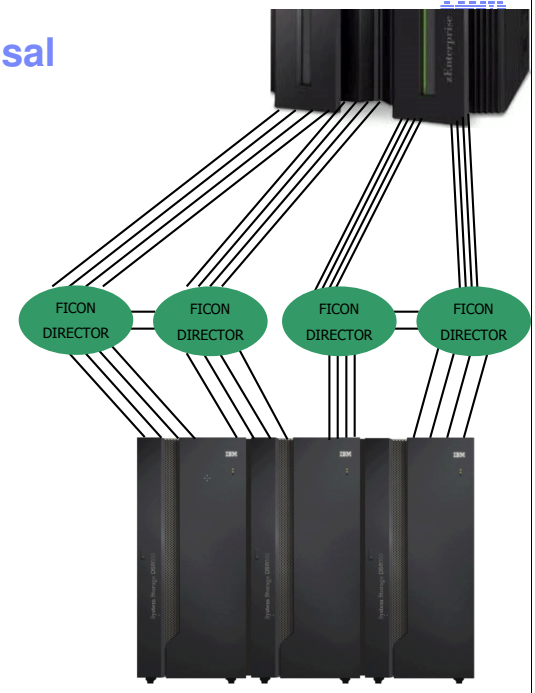
You can also control the available number ranges for control units and devices to suggest to the proposal processing how the device should be numbered. If free ranges exist within these policy suggestions, they will be used.

In the end, if you don’t like what numbers are proposed, you can change them in the work IODF to be what you think is a better numbering scheme.

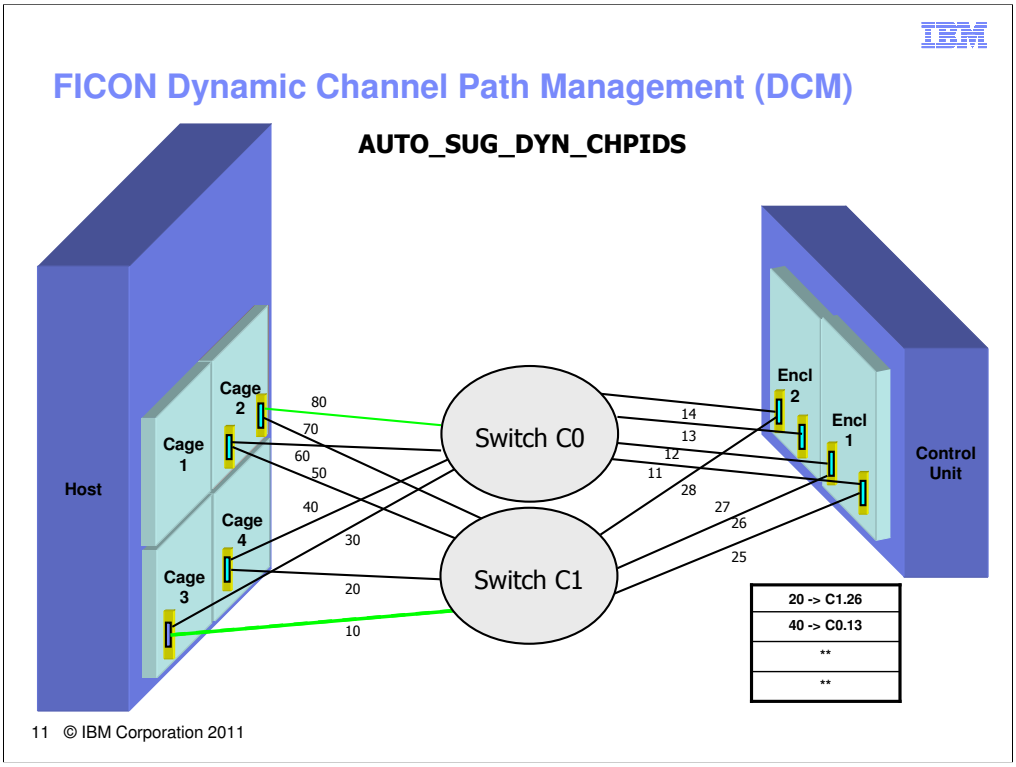
Controlling Path Proposal

- **AUTO_SUG_STAT_CHPIDS**
  - **1 – 8, indicating the number of static paths that should be proposed for new control units**
- **AUTO_SUG_DYN_CHPIDS**
  - **0-7, indicating the number of dynamic paths that should be proposed for new control units**
- **AUTO_SUG_STAT_CHPIDS + AUTO_SUG_DYN_CHPIDS <= 8**
- **Cascading Switches Are OK**

More on Path Proposal Processing Later…



Policy options allow you to indentify how many paths should be configured to newly discovered control units.



In the zDAC Policy, you specify how many static channels that should be configured for each new control unit.

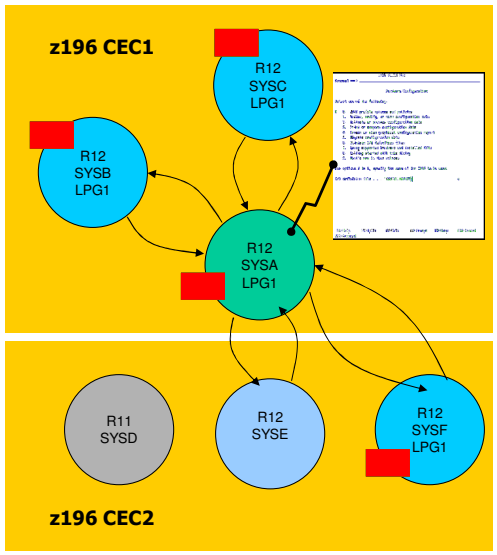
Static channels are chosen primarily based on availability characteristics. Performance is not a factor when paths are selected.

The AUTO_SUG_DYN_CHPIDS options allows for the specification of managed channels that can be added and removed to accommodate spikes in I/O workloads that cause temporary increases in I/O response times. The way this works in HCD is that channels can be identified as managed, and control units are defined with one or more "*" indicating managed channel path placeholders. As workloads change, z/OS monitors response times and, if necessary, will add and remove channels to/from those managed placeholder path positions. In the example, 2 static paths are configured (Channel 20 connects to switch C1 and switch port 26, channel 40 connects to switch C0 and switch port 13) and there are two paths identified as managed. z/OS could use channel 10 or channel 80 (managed channels shown in green) to connect to one of the shown ports if the need arises.

z/OS DCM can make path changes to help with both performance and availability. zDAC can propose configurations for availability. Together, the two can produce configurations that have continuous availability and performance in mind.

How Discovery Works – Fabric Discovery

- **Discovery is initiated using HCD or HCM**
- **Through policy, the scope of the discovery is defined**
- **Discovery devices are added to the target systems using dynamic I/O configuration changes**
- **To explore the connected fabrics, the devices are connected to channels using dynamic I/O configuration changes and I/O commands are performed to determine what is "out there."**
- **Discovered information is compared against the target IODF to determine what is new**
- **Information is organized and a list of discovered controllers is displayed to the user**



Use an LPGROUP to ensure that the discovery scope only includes those systems capable of performing the discovery (ie., systems running z/OS V1R12 or later and those running on z196 processors or later).

In order to explore the fabrics, temporary discovery devices are added to each discovery system and subsequently connected to channels to enable the discovery. These additions are done using dynamic I/O configuration changes. At least 1 system running z/OS V1R12 (or later) on each CEC involved in the discovery must have authority to make dynamic I/O changes on behalf of the processor. The temporary discovery device is connected to all switched FICON channels (one at a time) for the discovery of destination ports.

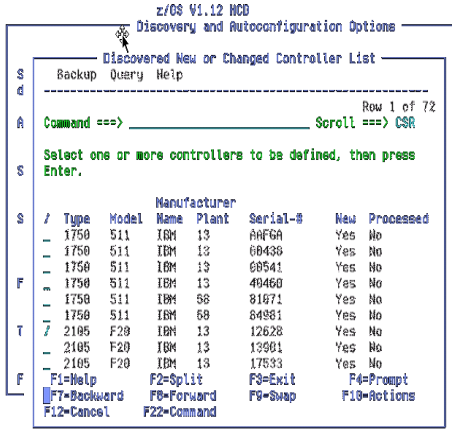
Fabric discovery is the discovery phase where all reachable destination ports are queried to determine what is there. First, an I/O command to the fabric name server is used to retrieve any destination port that was registered as a FICON channel. This is important because only those controllers (and processors) that register ports/channels as FICON channels will be returned. Check with your storage controller manufacturers to determine if their controllers will support this discovery capability.

Once all the destination ports are known, each one is queried to obtain a node descriptor. The node descriptor identifies the type of controller, model, manufacturer and serial number.

After all channels have been explored and all controllers have been discovered, the information is compared against the target work IODF to gain understanding of what is new and what is previously configured.

How Discovery Works – Fabric Discovery…

- **The controller list has a line for each discovered controller**
- **Information displayed is read from each controller**
- **New: Yes or No – set based on whether the controller has any control unit definitions within the discovery scope**
- **One or more controllers can be selected for Controller Discovery**



A list of controllers is then displayed to the user.

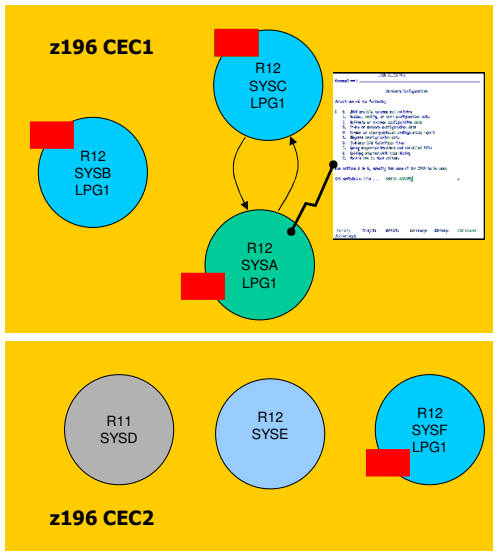
In order to make this list, a controller must be a tape or disk controller, must have properly registered with the fabric name server, and must have been reachable from all discovery systems.

Note that the new=yes/no column may indicate that a controller is new even though some definitions exist in the IODF. The yes/no indicator is correct for the systems within the scope of the discovery. For example, on the previous chart, controllers defined only to SYSD and SYSE (which are outside of the discovery scope specified by LPGROUP ***LPG1***) will appear as new.

One or more controllers can be selected for controller discovery. Depending on the discovery options selected, proposed definitions will either be displayed to the user or will be added to the target work IODF.

How Discovery Works – Controller Discovery

- **To explore the controller, the devices on a *single* system are connected to channels using dynamic I/O configuration changes and I/O commands are performed to determine what is "on there."**
- **Most I/O commands used have existed for a long time**
- **Discovered information is compared against the target IODF to determine what is new**
- **Paths are proposed for new control units**
- **New control units and devices are displayed to the user**



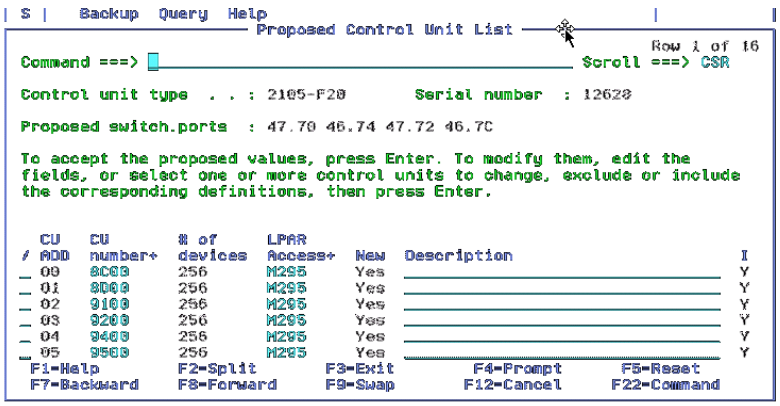
Controller discovery works with one controller at a time. If multiple controllers are selected, they are discovered serially.

Again, dynamic I/O configuration changes are made, this time on a single system where discovery of the selected controller(s) will take place.

If a selected controller is not new (ie., it has some devices configured on the discovery system), those existing devices may be able to be used for discovery, circumventing the need to make some or all of the necessary I/O configuration changes.

For any new control units, paths must be proposed. New control units will have paths proposed for each unique processor/channel subsystem that contains a discovery system.

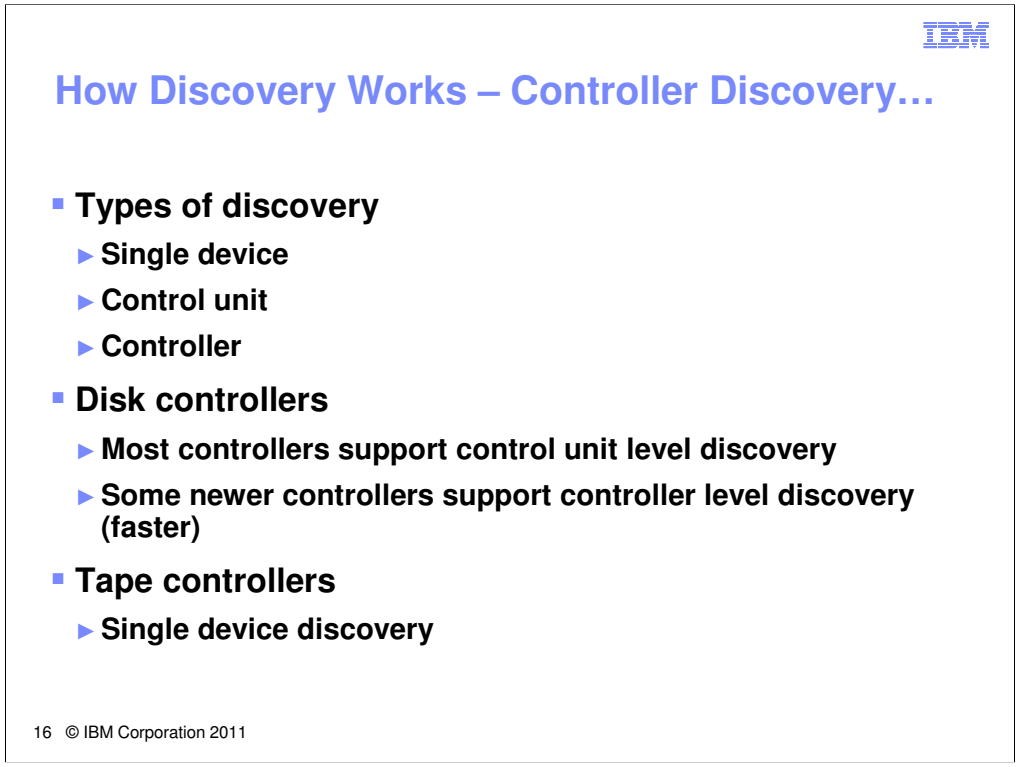
How Discovery Works – Controller Discovery…



Here, the list of discovered controllers is displayed.

On this panel, you can select a control unit to see more detail on the definition, including what paths were selected. If there are control units presented that you do not want to add to the IODF, you can exclude them on this panel.

Pressing enter will show proposed devices.



The time it takes to discover a controller depends on the controller type and the capabilities of the controller.

There are 3 types of discovery

-Single device discovery – in this mode, a device must be configured for each device to be discovered, and I/O commands must be performed to each device to determine if it is configured.

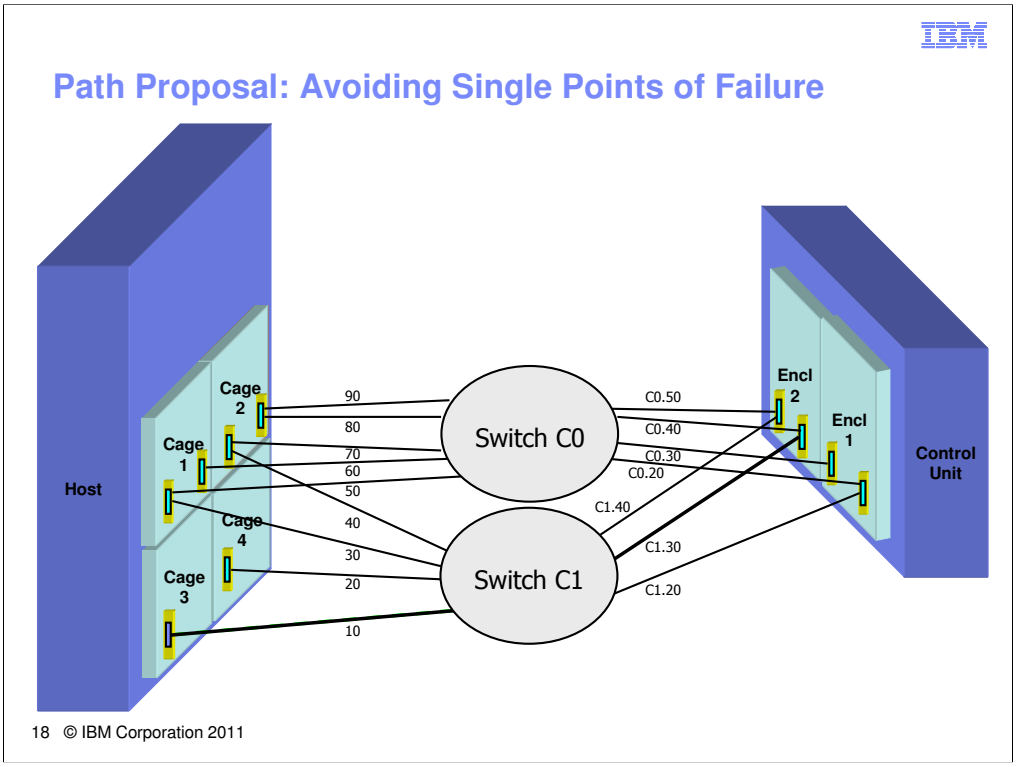
-Control unit discovery – in this mode, all devices attached to a control unit can be discovered using I/O commands delivered to a single device in the control unit

-Controller level discovery – in this mode, multiple control units and devices attached to them can be discovered using I/O commands delivered to a single device.

One thing to note is that during controller discovery, information is read from the devices and can automatically be configured in the IODF. Types of information discovered is control unit model, device type (ie.,base or alias disk devices), whether compaction is supported (tape), library ID (tape) and library port ID (tape).

Philosophy on Path Proposal

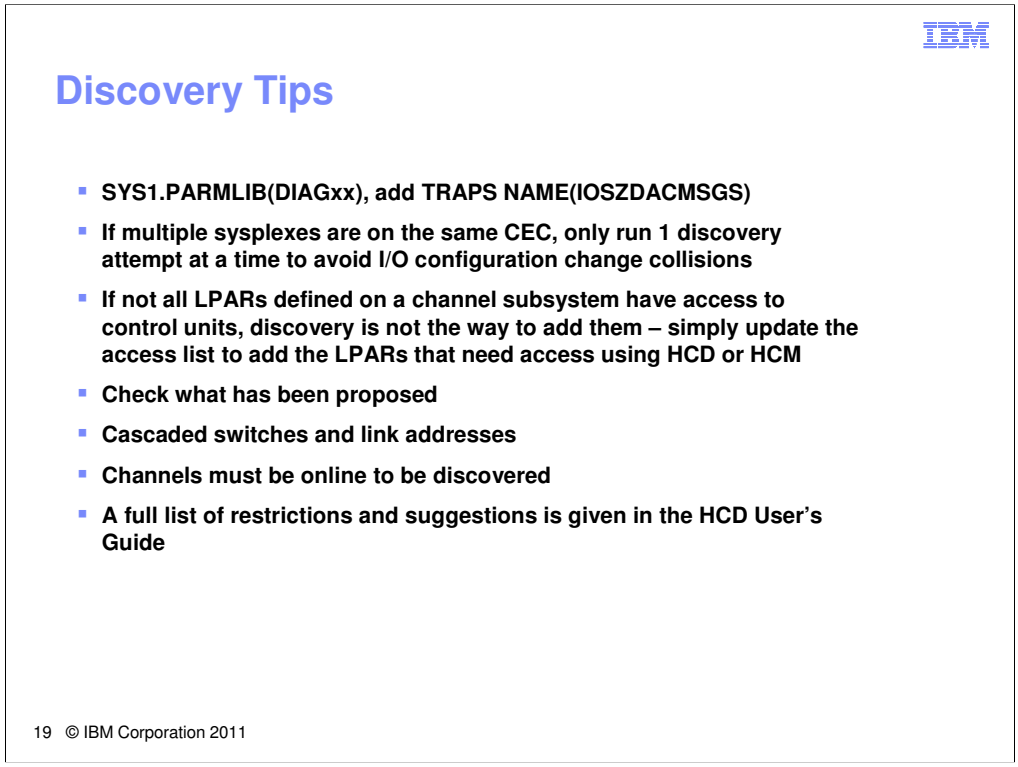
- Policy suggests # of paths to be selected (ie., 4 statics)
- The channels that can reach the controller are compared against each other to determine the best channels to use
  - Single points of failure (common hardware components)
    - Switch
    - Channel card
    - I/O cage
  - Current definitions (how many unit addresses and control units are currently configured to use the channel)
- Once a set of channels is selected, destination link addresses are chosen in a similar manner



This is kind of like a “find the hidden objects” picture. Look at the diagram and try to find why pairs of paths that would not be good choices.

For example, picking channels 80 & 90 as a pair would not be a good choice to get to this control unit because they share many common components. They are channels on the same adapter card, they share the same I/O cage, the same switch. A better pair would be channel 10 and channel 90, which do not share common components.

When a channel is selected, it carries with it a set of control unit ports that can be used to complete the path. Similar logic is applied when picking control unit ports. Path proposal would avoid selecting two destination ports that are on the same adapter card, same enclosure, etc.



SYS1.PARMLIB(DIAGxx) TRAPS NAME(IOSZDACMSGS) causes some diagnostic messages to be written to the SYSLOG, which can help avoid recreates if you need to contact IBM support. It also allows you to see some progress during discovery attempts.

In order to discover new controllers and control units, it is required that dynamic I/O configuration changes be made on each CEC hosting a discovery system. You want to make sure that conflicting changes are not performed simultaneously, which could adversely affect the discovery attempts. You should also avoid activating an IODF configuration change while discovery attempts are being made.

Another case where conflicts may arise in I/O configuration changes can occur when partition access lists are used to limit control unit access to a subset of LPARs on a channel subsystem. When a discovery attempt occurs on a partition that does not have access, attempts to add discovery devices could fail because other LPARs already have the control unit defined. The only way to add the control units to LPARs that don't have access is to remove the access list (granting all LPARs access to it on the channel subsystem), or grant access to the specific LPARs needing access.

You also will want to ensure that the devices are defined properly in the configuration. No esoterics are assigned to newly discovered devices, and defaults are taken for many device and control unit settings. One example is the LOCANY setting, which defaults to NO. This can be overridden using the HCD profile option 1 with the OS_PARM_DEFAULT = LOCANY,YES policy setting.

Architecture requires that link addresses be used consistently on a channel. If a channel has an existing path configured using only switch port addresses (a one-byte link address), cascading switch paths are not allowed to be defined on that channel. Using two-byte link address will allow for greater connectivity options (and will allow zDAC to propose cascaded paths).

Requirements

- **Systems running z/OS Version 1 Release 12**
- **z196 Processor**
- **LPAR authorized to make dynamic I/O configuration changes on each processor hosting a discovery system**
- **HCD or HCM user authorized to make dynamic I/O configuration changes (has UPDATE authority for MVS.ACTIVATE OPERCMDS resource)**

Just reaffirming the environment required for discovery and autoconfiguration:

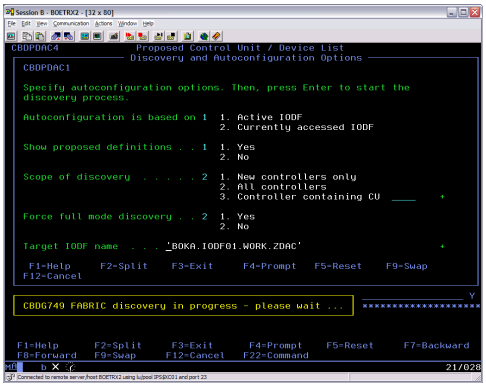
•z/OS Systems must be running V1R12 or later

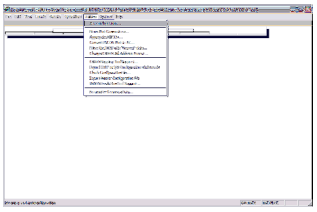
•Those systems must be running on a z196 processor or later

•At least 1 LPAR on each processor hosting discovery systems must be running z/OS V1R12 and must be capable of making dynamic I/O configuration changes

•The User must have authority to make those changes. This is the same resource (MVS.ACTIVATE) previously required to activate an IODF or IOCDS using HCD or HCM.

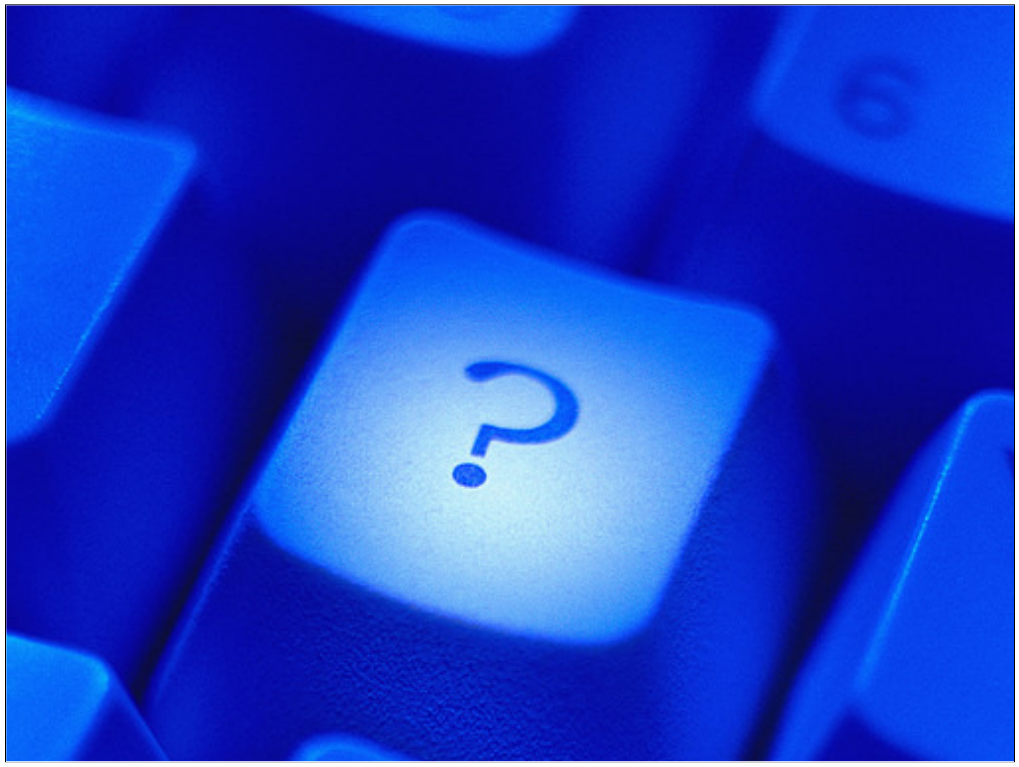
Demo





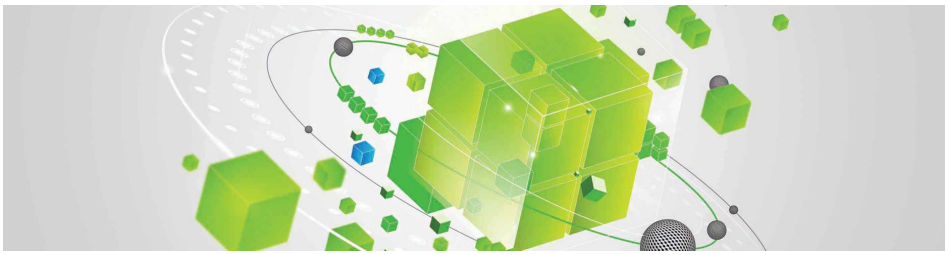
Reference Material

- z/OS V1R12.0 HCD User's Guide, SC33-7988
- z/OS V1R12.0 and z/VM V6R1.0 HCM User's Guide, SC33-7989
- IBM zEnterprise 196 Configuration Setup, SG24-7834
- z/OS Intelligent Resource Director, SG24-5952



Backup Screen Shots

If demo is not available…

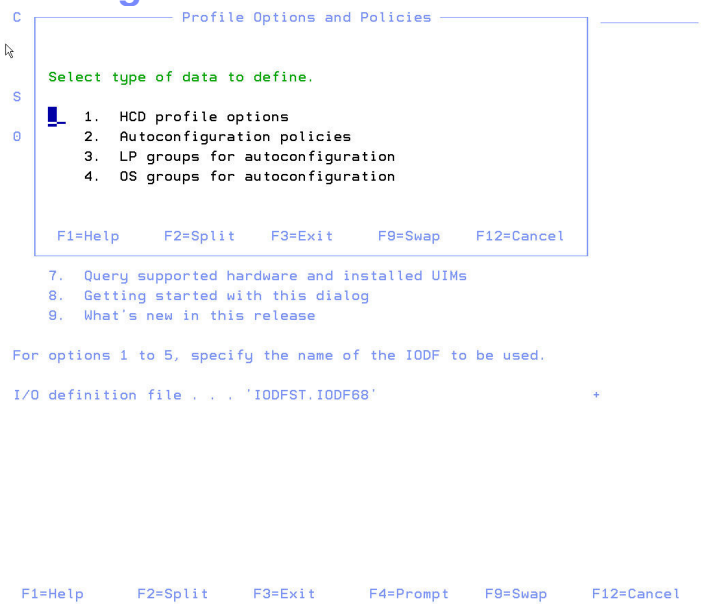


Discovering with HCD

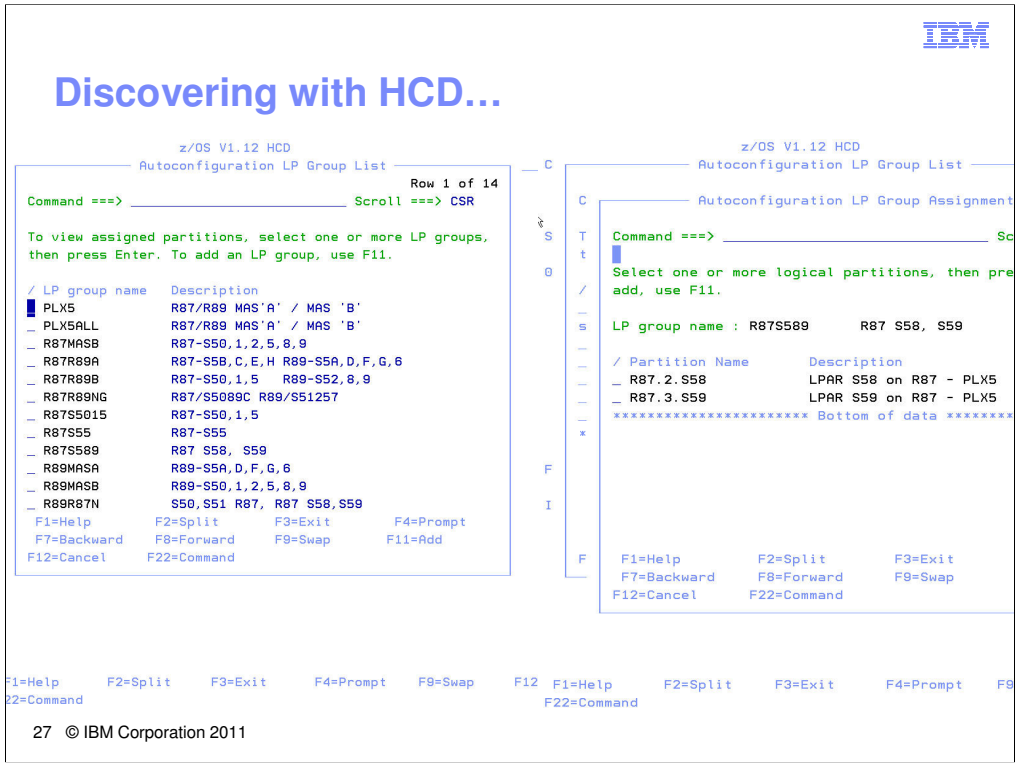
```
Command ===> ____________________________________________________________

                          Hardware Configuration

Select one of the following.

0  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 . . . 'IODFST.IODF68'                          +

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


Show here is the main HCD panel. Option 0 allows the user to establish policy information.

Discovering with HCD…



Here, 4 options are availble to the user. Discovery and Autoconfiguration processing can be tailored using options 2, 3 and 4. First, we will select option 3 to define which active LPARs will be in the discovery operation.

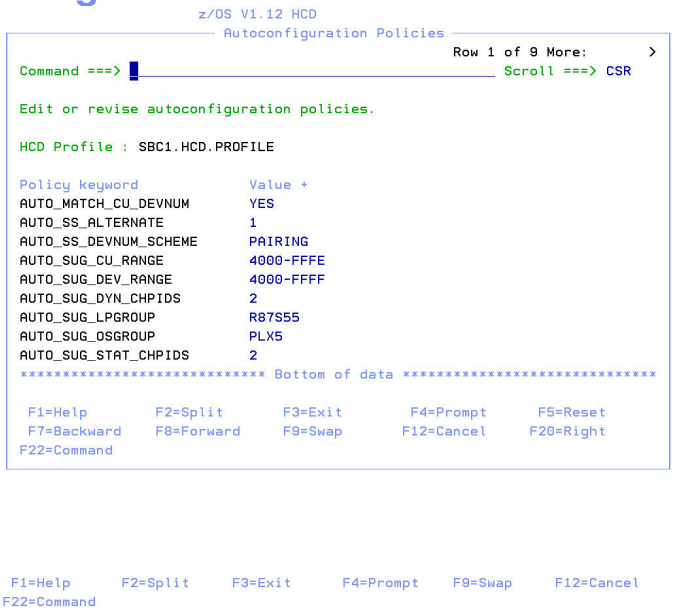


If a list of LPAR Groups already exists, you can select one and modify it. If none exist, LPAR groups can be defined using F11. Once an LPAR group is selected, navigating through the panels will allow partitions to be included in the LPAR group. Shown here are two panels, one containing the list of LPAR groups defined, and the other an LPGroup definition showing two LPARs in the group.

LPAR Groups can contain LPARs from many different processors. They will be used to define the scope of the discovery operation.

Likewise, 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. They can be created and modified using option 4 of the main policy panel.

Discovering with HCD…



Profile option 2 allows you to define discovery and autoconfiguration policy. You can control things such as device and control unit numbering, how many paths should be added to new control units, which subchannel set alias devices should be added to, and the LPAR group or OSGROUP that you wish to use to control the scope of the discovery.

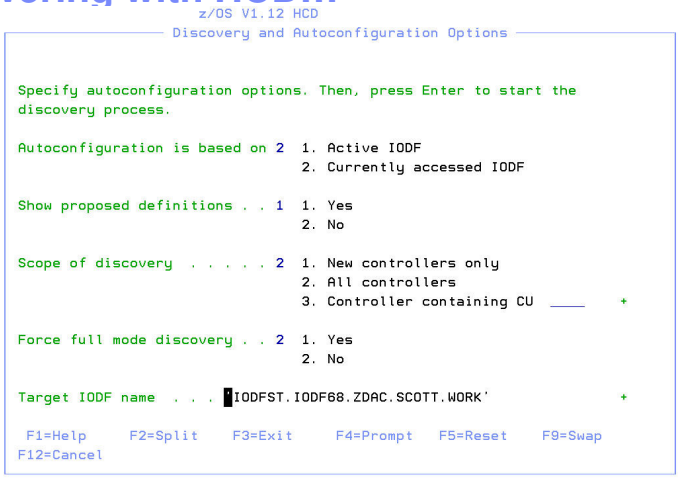
This information is stored in the user profile dataset.

Note here that 2 static channel paths and 2 dynamic (or managed) channel paths are requested, allowing DCM to manage paths to optimize performance for new control units.

Discovering with HCD…

Once the policy has been established, the Discovery and Autoconfiguration process can be started by selecting option 1 (Define, modify, or view configuration data) and then option 6 (Discovered new and changed control units and I/O devices).

Discovering with HCD...



Here, more discovery options are presented to give the user more control.

If the target work IODF is not yet created, it will be copied from the source IODF (either the active IODF or the currently accessed IODF, based on the first option selected).

You have the option of seeing what is proposed, or it can actually just be added directly to the target work IODF.

Once enter is pressed, the fabric discovery phase begins. Switch-attached FICON channels are explored to determine what controllers are reachable.

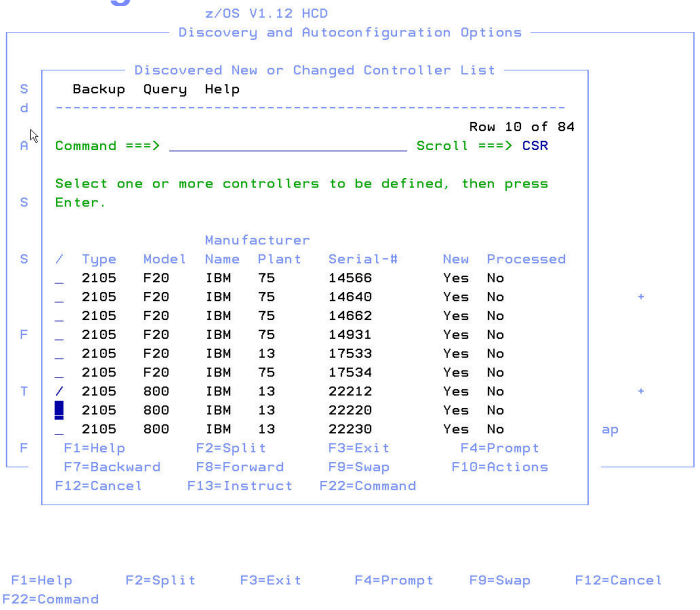
Discovering with HCD...

Once you press enter, a message is displayed indicating fabric discovery is in progress.

Discovery processing involves multiple address spaces. The TSO user waits on the foreign processes to complete. Pressing attention is allowed, but being a former TSO/E developer, it isn't something that I like.

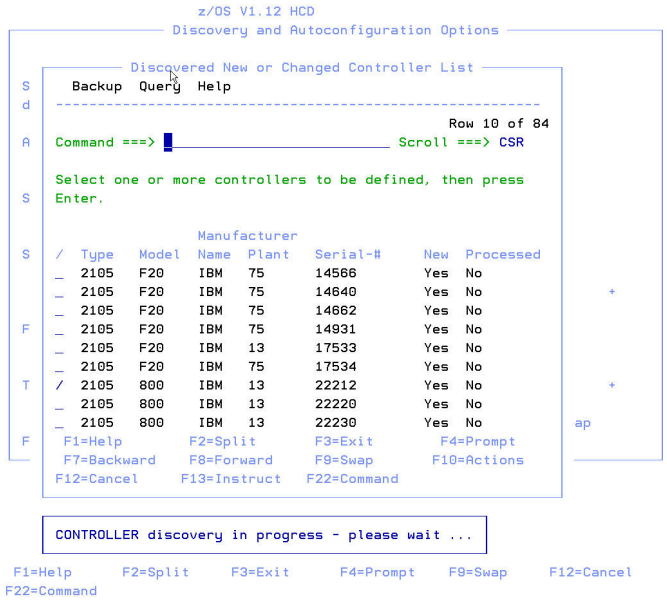
Fabric discovery times will vary depending on the number of channels that need to be connected and explored, and will also depend on how many switch ports need to be explored. The I/Os do not take very long, but the I/O configuration changes can take a second or two. Patience is always a good thing here. A guess at expected times would be 1 to 2 minutes. If it hurts to wait, consider running with the suggested DIAGxx TRAP and watch SYSLOG to see progress.

Discovering with HCD...



When fabric discovery is complete and successful, a list of controllers common to all discovery systems is displayed. One or more controllers can be selected for controller discovery.

Discovering with HCD…

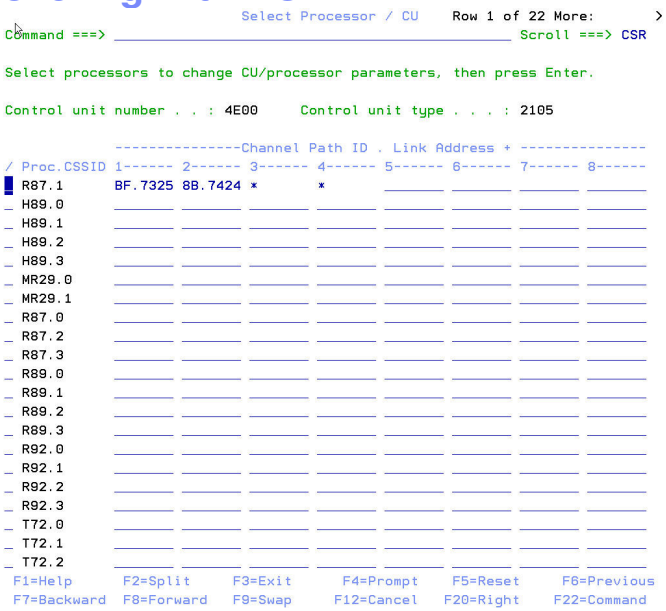


When fabric discovery is complete and successful, a list of controllers common to all discovery systems is displayed. One or more controllers can be selected for controller discovery. Patience please!

As was mentioned in the main presentation, the time it takes to do controller discovery depends on the capability of the controller. 2105 disk controllers have only 16 control units and typically support control unit discovery mode. 2107 disk controller that do not support the controller level discovery can take a long time, often several minutes if it contains a lot of control unit definitions. The DS8700 is a disk controller that does contain controller level discovery can discover an entire controller in seconds. And finally, tape controllers are relatively quick (usually less than a minute) even though they are discovered using single device discovery mode.

When controller discovery is complete and new control units and/or devices are found, the control unit list is displayed.

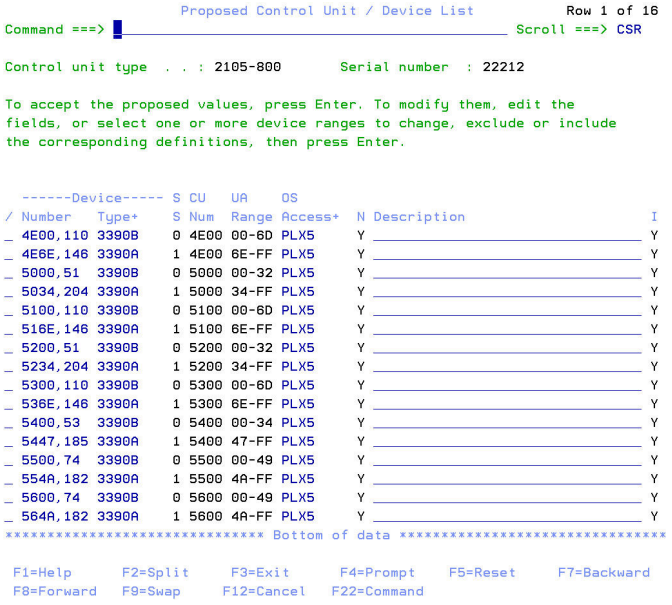
Discovering with HCD...



The selected control unit is displayed. You can see that my discovery system S55 runs on processor R87 and channel subsystem 1, and 2 static paths are proposed. Also note that 2 dynamic paths are proposed as well, denoted with '*'.

Here, you also see that both channels (BF and 8B) use 2 byte switch port addressing for destination ports.

Discovering with HCD…



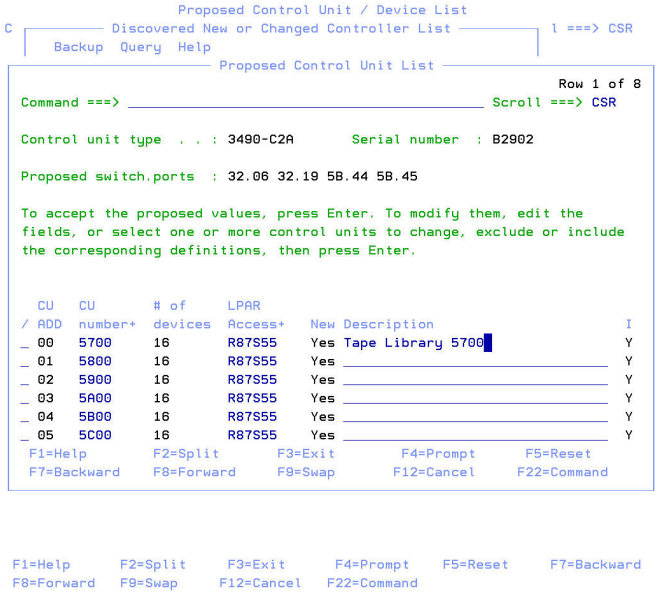
Here, the list of devices to be added to the target work IODF is displayed.

Pressing enter will accept the proposed device numbers. Pressing F3 will not accept the proposed devices.

Discovering with HCD…

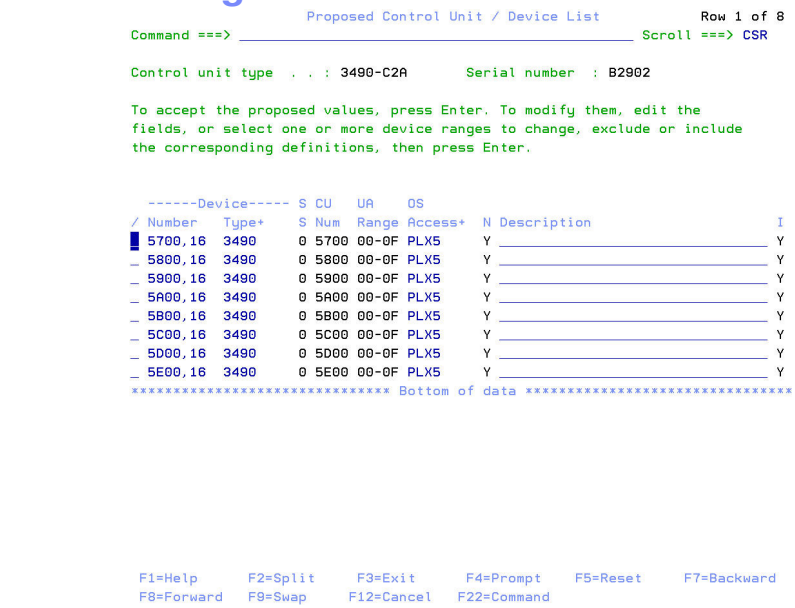
You can select a device and look at its proposed definition. Note that all disk devices are defined as 3390B or 3390A depending on its configuration on the controller. The discovery processing will never propose a 3390 or 3380 device type. If this is what you want, you will need to change this manually.

Discovering with HCD…



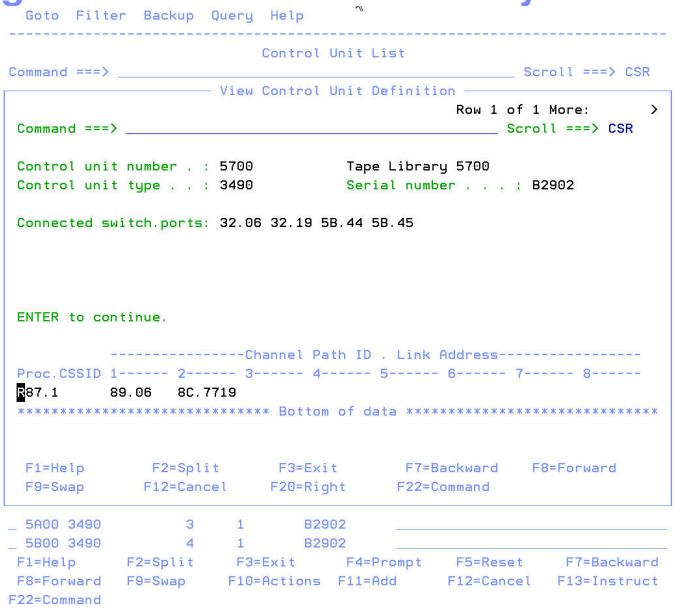
Shown here is a control unit list that was displayed when I selected a tape controller for discovery. Notice that you can edit CU numbers and description fields. You can also exclude control units by using the “e” command.

Discovering with HCD...



Show here is a list of tape devices proposed.

Looking in the IODF after Discovery

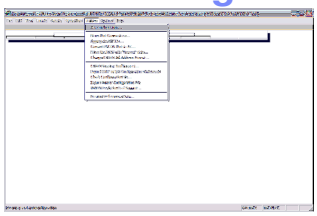


Using HCD option 4 (Control Units), you can select one of the discovered controllers and inspect the control unit definition.

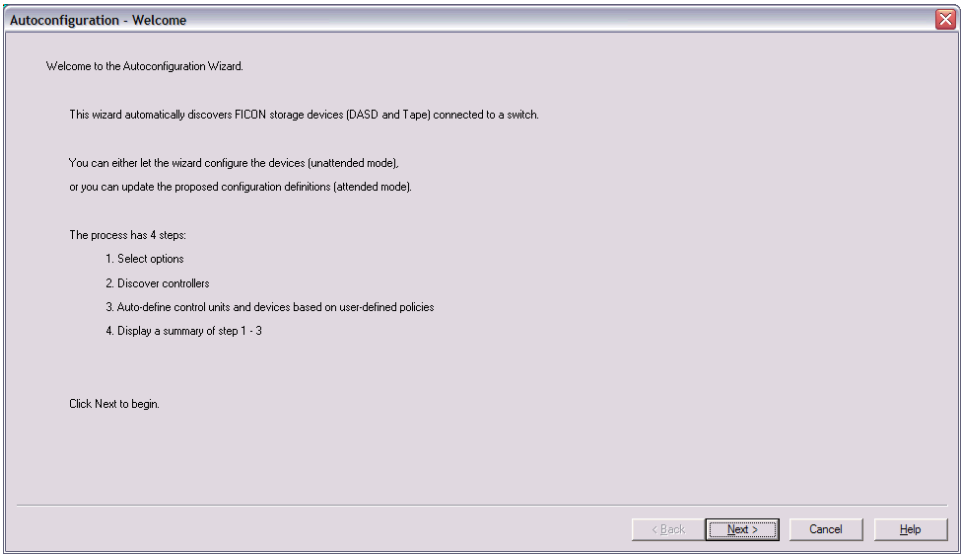
Looking in the IODF after Discovery

Show here are the device parameters and features. Some of these use defaults defined in the UIM for the devices, and some are discovered using I/O commands to the devices.

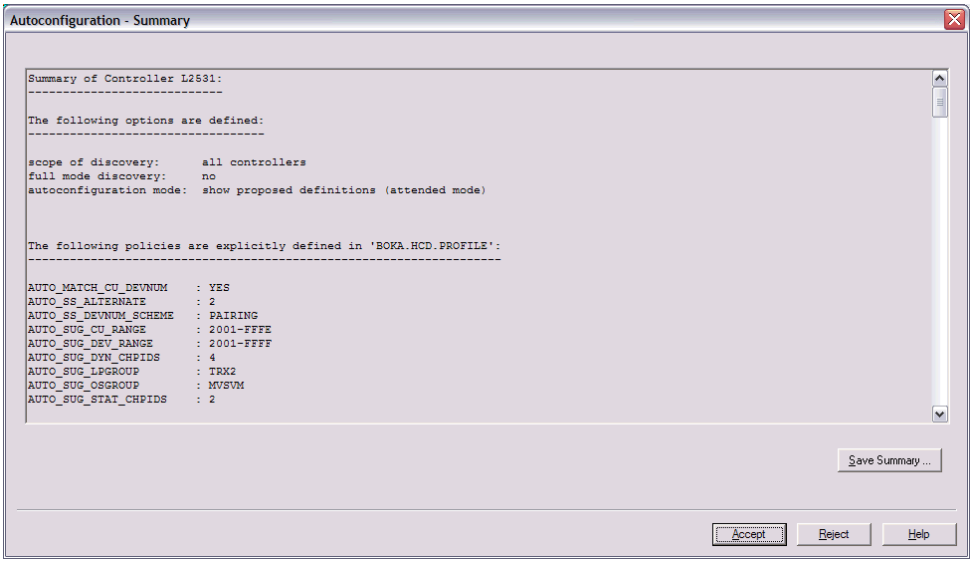
Discovering with HCM



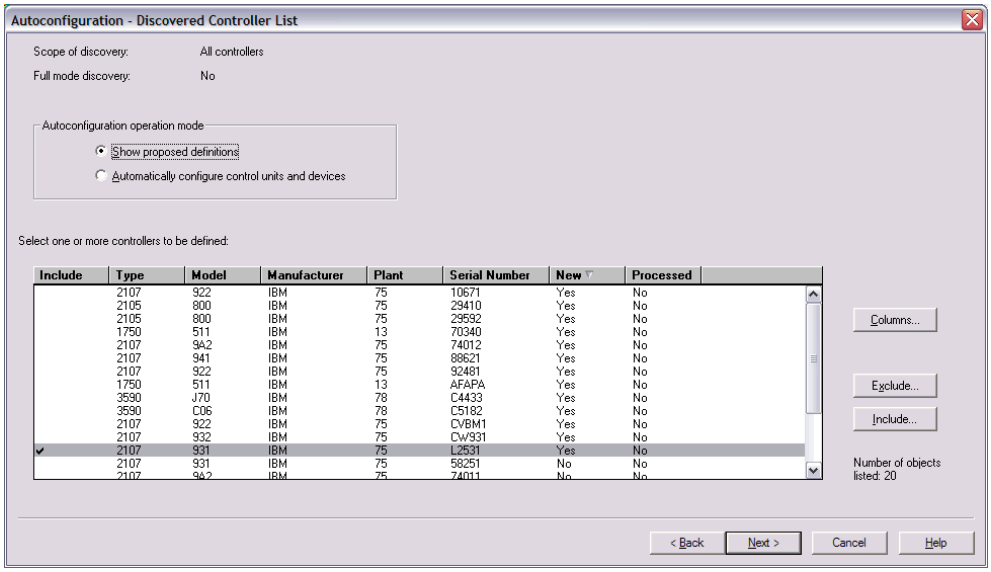
Discovering with HCM…



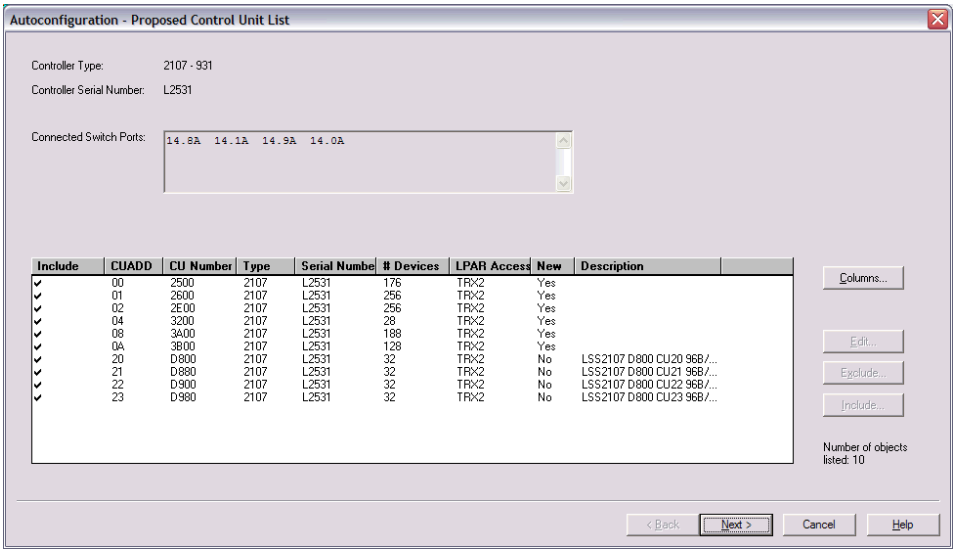
Discovering with HCM…



Discovering with HCM…



Discovering with HCM…



Discovering with HCM…