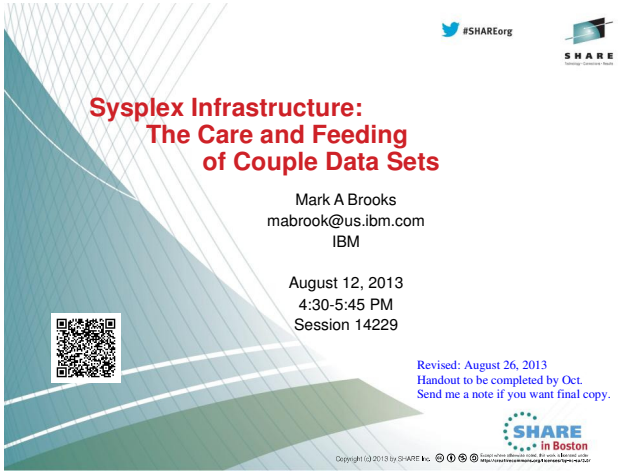


This session provides a comprehensive overview and detailed discussion of couple data sets (CDS). The speaker will cover basics such as: What is a CDS? Why do I need one? How is it used? How do I create one? How do I get the sysplex to use it? He will discuss in detail the sysplex couple data set, the various function couple data sets, and explain terminology such as "primary CDS", "alternate CDS", "PSWITCH", "active policy", "administrative policy", "format utility", and "data utility".

There will also be a discussion of best practices and "real world" mistakes seen by the XCF level 2 service team with an eye towards helping you avoid them in your shop (and how to recover if possible).

Those that are new to sysplex should walk away with a firm grasp of all that is needed to confidently deal with couple data sets in their sysplex. Those with experience will find value as well -- even if most of the basics are well known -- as they should walk away with a more detailed understanding that should enable them to better ensure that couple data sets in their sysplex are configured and managed in a way that avoids unnecessary risk. After all, mistakes with a CDS can lead to a sysplex wide outage!

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM®	MQSeries®	S/390®	z9®	IBM® (logo)
ibm.com®	MVS™	Service Request Manager®	z10™	AIX® BladeCenter®
CICS®	OS/390®	Sysplex Timer®	z/Architecture®	DataPower®
CICSPlex®	Parallel Sysplex®	System z®	zEnterprise™	DS4000®
DB2®	Processor Resource/Systems Manager™	System z9®	z/OS®	DS6000™
eServer™	PR/SM™	System z10®	z/VM®	DS8000®
ESCON®	RACF®	System/390®	z/VSE®	POWER7®
FICON®	Redbooks®	Tivoli®	zSeries®	ProtecTIER®
GDPS®	Resource Measurement Facility™	VTAM®	zEC12™	Rational®
HyperSwap®	RETAIN®	WebSphere®	Flash Express®	System Storage®
IMS™	GDPS®			System x®
IMS/ESA®	Geographically Dispersed Parallel Sysplex™			XIV®

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 there from.
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.
InfiniBand is a trademark and service mark of the InfiniBand Trade Association.
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.

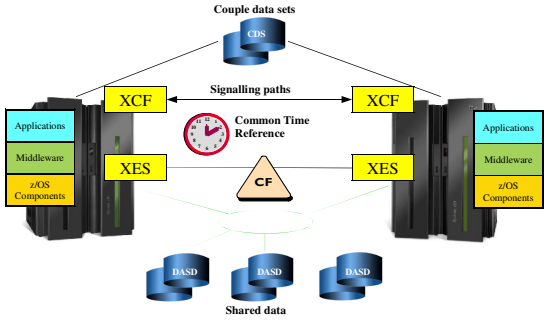
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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.

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

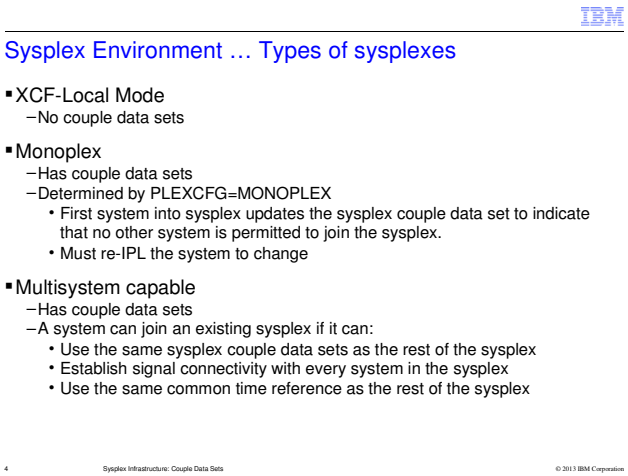
Sysplex environment



The context for this presentation is generally a sysplex consisting of two or more z/OS images. The images could reside on separate CECs as suggested in the diagram, or could reside in different LPARs on the same CEC. Thus for us, a sysplex consists of:

- Two or more z/OS images
- Using the same sysplex couple data set(s)
- Connected by XCF signalling paths for communication
- With a common time reference (STP – Server Time Protocol)
- With an optional coupling facility (CF).

XCF is the component of z/OS that manages the base sysplex infrastructure (couple data sets and signalling paths). XES is the component of z/OS that manages the coupling facility. A sysplex with a coupling facility is called a “parallel sysplex”. Both XCF and XES provide programming interfaces that allow applications to exploit the sysplex environment. Such applications will typically have one or more instances running on one or more z/OS images. These instances cooperate with one another to achieve their intended function. Applications may be written directly to the XCF/XES interfaces but are often written to exploit IBM or OEM middleware that in turn exploit the XCF/XES interfaces. In general these mulitsystem applications need to access common data that will reside on shared DASD. A variety of techniques and protocols are used to maintain the integrity of the shared data.



XCF-Local mode is a single system sysplex that does not have a sysplex couple data set. With no sysplex couple data set, the system cannot have function data sets either and that implies it will not be able to make use of a coupling facility. XCF signalling paths will not be used even if they are defined. To IPL into XCF-Local mode, simply use a COUPLExx parmlib member that does not define a sysplex couple data set. This mode could be useful if you need to run jobs to create data sets or parmlib members prior to IPLing into a “real” sysplex.

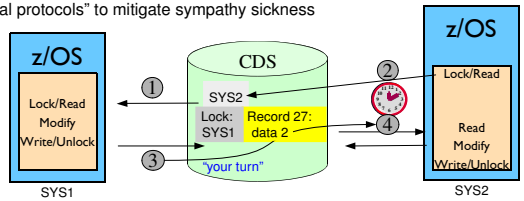
A **monoplex** is another form of a single system sysplex. Unlike XCF-Local mode, a monoplex does have a sysplex couple data set. Thus it can use function couple data sets and could exploit a coupling facility. If you IPL a system with a sysplex couple data set and never IPL any other system into that 1-system sysplex, you would in some sense have a monoplex. However, there is nothing to prevent some other system from joining that 1-system sysplex. If you truly need to prevent that possibility, you code the PLEXCFG parameter to specify MONOPLEX. That parameter causes XCF to update the sysplex couple data set to indicate that no other system is permitted to join the sysplex. If a system were to try, it will see the monoplex indication in the sysplex couple data set and refuse to join the sysplex. A reIPL of the “sysplex” clears it.

To participate in a **multisystem sysplex**, a system must have access to the sysplex couple data sets, the same common time reference being used by the other systems in the sysplex, and be able to establish communication with every other system in the sysplex via XCF signalling paths. If a system cannot maintain these conditions it must be removed from the sysplex because it will not be able to effectively participate in the sysplex. Removing a system from the sysplex causes it to wait-state.

We will later discuss function couple data sets. At this point, we note that a system might be able to IPL into a sysplex even if it cannot get access to the same function couple data sets that are already in use by the sysplex. If so, the system will not be able to use the relevant functions. However, there may be cases where the system will not make it into the sysplex. For example, if the system is to run in GRS-Star mode but cannot get access to the Coupling Facility Resource Manager (CFRM) couple data set, it will be wait-stated by the global resource serialization service (GRS).

Couple Data Sets (CDS)

- Provide means to harden data and share it between systems in the sysplex under serialization
- Accessed via XCF channel programs and protocols
  - Typical usage
    - Lock record and read content into storage
    - Modify in-store copy
    - Write modified content to CDS and unlock record
  - "Lock steal protocols" to mitigate sympathy sickness



Generically, a couple data set provides a central shared repository of data that needs to be visible to every system in the sysplex. The CDS and all accesses to the data therein are managed by XCF. None of the traditional access methods with which you might be familiar are used for these data sets. The data set is only accessed via XCF channel programs. There are several reasons. Some relate to concerns over software layering wherein we want XCF to be sufficiently low in the stack that we can maximize the opportunity to enable sysplex applications. Others relate to history. At the dawn of sysplex we sometimes had to invent and implement services that were not otherwise available at the time. Regardless, XCF continues to manage these data sets and control the protocols used to access them.

The slide depicts a typical protocol for a CDS.

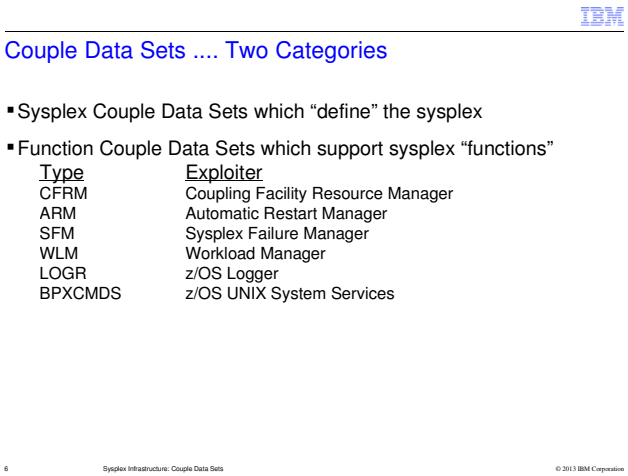
1) System SYS1 initiates a serialized read to fetch record 27 from the designated CDS. The channel program updates the record to indicate that SYS1 has locked the record. A copy of the data in record 27 is fetched into storage. SYS1 modifies the in-storage copy of the data.

2) While the record is locked, SYS2 performs a serialized read of the same record. Since SYS1 has locked the record, the read fails. SYS2 updates the lock in the record to indicate that it has an interest in the record.

3) System SYS1 writes the modified data back out to the CDS, releasing the lock. It notices that SYS2 had an interest in the record, and sends a signal to SYS2 to indicate that the record is now available. Had there been more than one system with an interest, SYS1 would have selected a system to be "next". Normally this is the order in which interest is expressed.

4) There could be significant delay since step (2). SYS1 might take a long time to get to the point where it can release the lock and/or the signal might be delayed as well. When system SYS2 receives the "your turn" signal, it reads the new record content, updates the in-storage copy, writes the modified data to the CDS and releases its lock.

To avoid sympathy sickness, a system can "steal" the lock out from under a system that has locked a record but does not seem to be making progress. If SYS2 had stolen the lock from SYS1, the write and unlock in step (3) would have failed. SYS1 would then have start over with a new serialized read of the record, rework its modifications based on the current content, and try again to write/unlock the record to harden the change.



A sysplex requires a **sysplex couple data set** (Sysplex CDS) to store information about its systems, the XCF groups and members running in the sysplex, and general status information. In essence, the sysplex CDS defines the sysplex since it is the repository that maintains the status of every system in the sysplex.

Depending on the various functions that you configure to help manage resources and workload for the sysplex, you might need to define additional **function couple data sets** (function CDS) for their use:

•The coupling facility resource management (CFRM) couple data set holds the CFRM policy that defines how z/OS is to manage your coupling facility resources.

•The automatic restart management (ARM) couple data set holds the policy that defines how z/OS is to manage restarts for specific batch jobs and started tasks that are registered as elements of automatic restart management.

•The sysplex failure management (SFM) couple data set holds the SFM policy, which allows you to define how system failures, signaling connectivity failures, member failures, connector failures, and PR/SM reconfiguration actions are to be managed.

•The workload management (WLM) couple data set holds the WLM policy that defines service goals for workloads running in the sysplex.

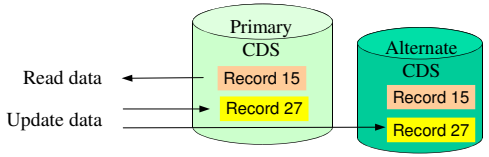
•The system logger (LOGR) couple data set holds the policy that allows you to define log stream or structure definitions.

•The z/OS UNIX System Services (BPXMCDS) couple data set contains information about mounted file systems and their use within the sysplex.

Sometimes my terminology can be confusing. On this slide I suggest that there are two types of CDS, the sysplex CDS and function CDS. But there is a TYPE keyword typically used in COUPLExx parmlib members or SETXCF commands to indicate what type of CDS is to be manipulated. In that context, TYPE could be any one of the seven types of couple data sets: SYSPLEX, CFRM, ARM, SFM, WLM, LOGR, or BPXMCDS.

Couple Data Sets … Primary and Alternate

- Normally run with both Primary and Alternate CDS
  - Read requests directed to primary
  - Update requests written first to primary, then to alternate
    - Both must complete for I/O request to finish
- Sysplex automatically switches to alternate if primary fails
- **Loss of both primary and alternate can be disastrous**
  - Wait-state of every system in the sysplex, or
  - Significant loss of sysplex function



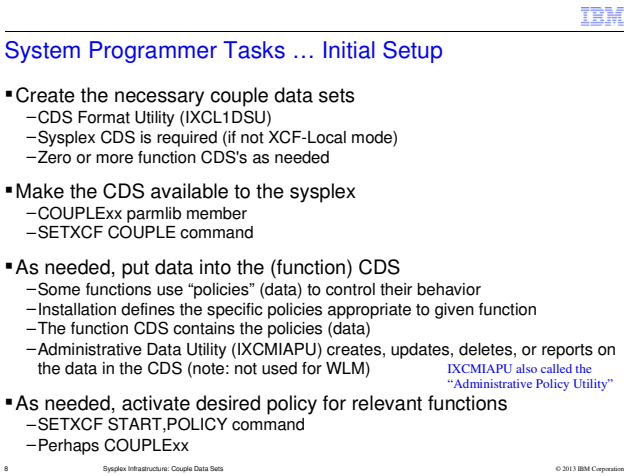
The couple data set represents a single point of failure for the function that needs it. The function may cease to operate if its CDS is not accessible. For the sysplex CDS, that means every system that loses access will wait state. Same goes for CFRM. That is, loss of the sysplex CDS or the CFRM CDS could cause a sysplex outage. For the remaining functions (ARM, SFM, WLM, LOGR, BPXMCDS), the sysplex will experience some loss of functionality. You may not lose the sysplex, but there could be significant impacts. In large part the impact will be determined by nature of the function and the circumstances that exist while the CDS is inaccessible. Some examples:

•ARM: If none of the elements being monitored by ARM fail, lack of the ARM function will not be noticed. On the other hand, if your automation package goes down and you were relying on ARM to restart it, your operations staff might have their hands full until such time as someone figures out that the automation package needs to be restarted manually (and they probably ought to fix the ARM CDS while they're at it).

•Unix System Services: Loss of the CDS would imply that one or more of the following file system functions could be delayed until access to the CDS is restored: file system initialization, mount processing, unmount processing or partition recovery.

To help mitigate the loss of function and/or system failures, you should always run with two of each type of couple data set, a primary CDS and an alternate CDS. If the alternate fails, XCF removes it from service and you will be running with just the primary (and a single point of failure). If the primary CDS fails, XCF removes it from service and automatically switches over to using the alternate. The alternate takes on the role of being the primary CDS (and you have a single point of failure since there is now no alternate).

When an alternate CDS is brought into service, XCF synchronizes the data sets by copying all the data from the primary into the alternate. Once synchronized, XCF keeps them in sync. As depicted in the slide (for record 27), any update written to the primary CDS will also be written to the alternate CDS. A pure read operation is directed only to the primary CDS (record 15 in the slide). When performing an update, the I/O request is not deemed to have completed until the updated data is written to both the primary and the alternate.



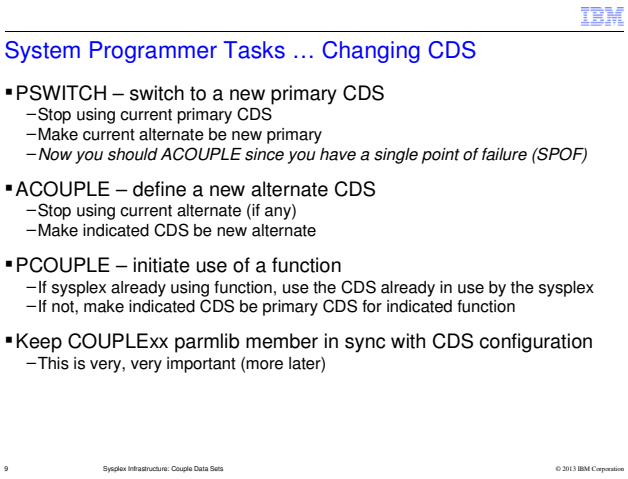
The slide summarizes key system programmer tasks required to set up the various couple data sets.

Create CDS: All CDS are created using the format utility IXCL1DSU found in SYS1.MIGLIB. The utility always creates a new data set and then initializes records in the data set appropriate to the type of CDS. Except for a system that IPLs in XCF-Local mode (which by definition does not have a CDS), a sysplex must have a sysplex couple data set. The function CDS are optional. If you have a base sysplex (no coupling facility), you do not need the CFRM CDS. If you are truly trying to garner as much benefit as possible from your sysplex investment, you will likely use every type of function CDS.

Make CDS Available to Sysplex: The CDS are defined in the COUPLExx parmlib specified when a system IPLs into a sysplex (we shall later see that things are a bit more complicated). After IPL, use the SETXCF command to make a CDS available to the sysplex dynamically. If a particular type of CDS is in use anywhere in the sysplex, every system that exploits that particular type must be using the same set of CDS. For a given type of CDS, at most two CDS can be in service. Thus the CDS configuration for a given type will either be the a primary CDS alone, or a primary and alternate CDS.

As Needed, Put Data into Function CDS: For SYSPLEX and BPXMCDS, the CDS is populated with data as the system performs work. You need not explicitly put data into these. For the other types of CDS (CFRM, ARM, SFM, WLM, and LOGR), you must explicitly put data into the CDS. In general, these CDS contain policies that define how you want the function to behave. Use the administrative data utility IXCMIAPU found in SYS1.PARMLIB to put data into the CDS. IXCMIAPU is also called the administrative policy utility. WLM is an exception in that its CDS content is created using a WLM application, not IXCMIAPU. In general, the systems making use of any CDS will store data in that CDS as they perform tasks related to the function. Thus most function CDS will contain a mixture of static data determined by the installation, as well as dynamic data stored by the sysplex at run time.

In general, you can either make the formatted CDS available to the sysplex and then put policy data into it, or you can put policy data into a formatted CDS and then make it available to the sysplex. There may be circumstances that require one particular order. Installations running GRS-Star mode, for example, often have situations that prescribe a particular order with respect to the CFRM CDS.
Activate desired policy: Most function CDS contain one or more administrative policies. A policy must be activated (started) for it to be used. The SETXCF START,POLICY command is used to start a policy. Under certain circumstances, the CFRMPOL keyword in the COUPLExx parmlib might be needed to activate a CFRM policy during IPL before the system can accept commands.



After the initial setup, there will come a time when changes need to be made. This slide summarizes the key concepts related to making changes to the CDS configuration. Applicable commands:

SETXCF COUPLE,TYPE=*typename,*PSWITCH
SETXCF COUPLE,TYPE=*typename,*ACOUPLE=(*cdsname,volser*)
SETXCF COUPLE,TYPE=*typename,P*COUPLE=(*cdsname,volser*)

PSWITCH causes the sysplex to switch to a different primary CDS. The sysplex will stop using the current primary CDS and begin using the current alternate CDS as the primary. Note that there must be an alternate CDS already in use for there to be a PSWITCH. If not, you would need to ACOUPLE in an alternate CDS before initiating the PSWITCH. You cannot switch to an arbitrary CDS, it has to be the current alternate.

ACOUPLE is used to bring a new alternate CDS into service. If an alternate CDS of the indicated type is currently in use, it is first removed from service. The designated CDS is then brought into service as the alternate CDS. After all the systems in the sysplex agree to use the CDS as the alternate, it will be synchronized with the primary CDS. During the synchronization, XCF copies the content of the primary CDS into the alternate CDS. Once synchronized, subsequent updates will be written to both the primary and the alternate.

PCOUPLE is used to start use of a function on a system that is not currently using the function. If the designated function (typename) is not in use anywhere in the sysplex, the designated CDS is brought into service as a primary CDS for that function (in most cases you will then need to start a policy). If the function is already in use anywhere in the sysplex, the CDS name specified on the SETXCF command is ignored. Instead, the system will use the same function CDS already in use by the rest of the sysplex. All systems making use of a function must at all times use the same CDS.

Contrast this with ACOUPLE which removes the current alternate and starts using the designated CDS as the new alternate. PCOUPLE never removes the current primary CDS and it only brings the designated CDS into service if the designated function is not being used anywhere in the sysplex.

System Programmer Tasks … Changing Policies

- Create, update, or delete an administrative policy
- Activate a new or changed policy
- Stop using a policy

- All policy based functions have similar concepts, but there are differences to be understood
  - CFRM, ARM, SFM
  - LOGR
  - WLM

Policies are not applicable to SYSPLEX or BPXMCDS

This slide summarizes the key concepts related to making changes to policies for the policy based functions: CFRM, ARM, SFM, LOGR, and WLM. Conceptually, all of the functions are quite similar. One or more policies are created, updated, or deleted; a specific policy is chosen to be applied to the sysplex; perhaps a policy is stopped (to disable the function or revert to some default behavior). The specifics of how this is done can vary for each function.

CFRM, ARM, SFM: These functions have administrative policies and an active policy. Use the administrative data utility IXCMIAPU found in SYS1.MIGLIB to add, update, or delete administrative policy data in a formatted CDS. The maximum number of administrative policies is determined by the number specified on the POLICY control statement in the couple data set format utility when the CDS was created. One of the administrative policies is activated for use in the sysplex by issuing:

SETXCF START,POLICY,TYPE=*typename*,POLNAME=*policyname*

Once a policy is started, it remains in effect until a new one is started or until it is stopped:

SETXCF STOP,POLICY,TYPE=*typename*

If you update the administrative copy of the active policy, you must start the policy again to have those changes take affect. If stop a policy, the function is disabled. CFRM will not completely shut down until all the CF structures are deleted.

LOGR: The LOGR function has but one policy which is always active. The LOGR policy is implicitly active when the LOGR CDS is placed into service. The policy cannot be stopped. IXCMIAPU is used to update the LOGR policy. Your changes are effective immediately. Some changes could be rejected if they are not appropriate for a logstream that is currently in use. In some sense, you use IXCMIAPU to make incremental changes to your LOGR policy. For example, you might create or delete logstreams definitions as needed. Contrast this with CFRM, ARM, and SFM where IXCMIAPU updates the entire administrative policy.

WLM: Neither IXCMIAPU nor SETXCF POLICY commands are used for WLM policy information. All activities related to WLM are accomplished through an ISPF application or a z/OSMF task. See *z/OS MVS Planning: Workload Management* for a description. Overview: WLM service definitions are created in traditional MVS data sets. A service definition is installed in the WLM CDS to activate it. The service definition has an implicit base policy. It may contain other service policies that can be started as well.

System Programmer Tasks … Related Concerns

- CDS Placement
  - Availability
  - Performance
  - Failure isolation
- CDS Failures
  - Avoiding single points of failure
  - In conjunction with unresponsive/failed systems
- Sysplex IPL and Disaster Recovery
  - What CDS's to use
- Various Risks
  - Mirroring of CDS
  - Using an old CDS copy

Although most of the CDS are accessed relatively infrequently, there may be situations where significant amounts of activity will occur. For example, the CFRM CDS is subject to significant spikes when a system is removed from the sysplex or when a coupling facility fails. Regardless of the amount of I/O activity, it is important that all I/O requests to a CDS to complete in a timely manner. The impact of I/O delays might be isolated to a single system. However, since a CDS is involved in sysplex wide protocols, there is a good chance the I/O delay will induce sympathy sickness. For example, CFRM CDS I/O delays encountered while recovering from a CF failure (for example) would have a direct impact on the amount of time it takes your applications to be restored to service. In the worst case, the I/O delay might cause XCF to remove the CDS from service. So we want to ensure that the CDS is always accessible with good performance.

If a CDS should fail, it will be removed from service. So we want to ensure that the primary and alternate CDS are failure isolated from each other. Otherwise, the failure that led to removal of one CDS could lead to the removal of the other. If both CDS are lost the relevant function is lost as well. In the case of the sysplex CDS or the CFRM CDS, there will be a sysplex outage. So be sure to run the z/OS Health Checker. The XCF_CDS_SEPARATION check will warn you if your primary and alternate CDS are not failure isolated from each other.

While a CDS is being removed from service, it is quiesced for all I/O activity. Thus removal of a sysplex CDS can be a challenging if a system fails during the removal. In particular, deadlocks can arise. We will later discuss this scenario in detail and the steps required to make progress.

Whenever you IPL a sysplex, it is important that you do so with the right instances of the various CDS. When reIPLing a sysplex, you should always use the CDS that were in use when the sysplex went down. When IPLing a sysplex at a DR site, you should always use freshly formatted CDS (except possibly for the LOGR CDS). Using a copy of a CDS or using a CDS that was previously in use by some other sysplex is a recipe for disaster. Many a customer has lost their production sysplex when doing a DR test. So we'll want to have a very detailed understanding of the issues behind this concern.

Thus DASD mirroring of a CDS or use of any other residual copy of a CDS should immediately raise alarms. You really, really need to be careful!

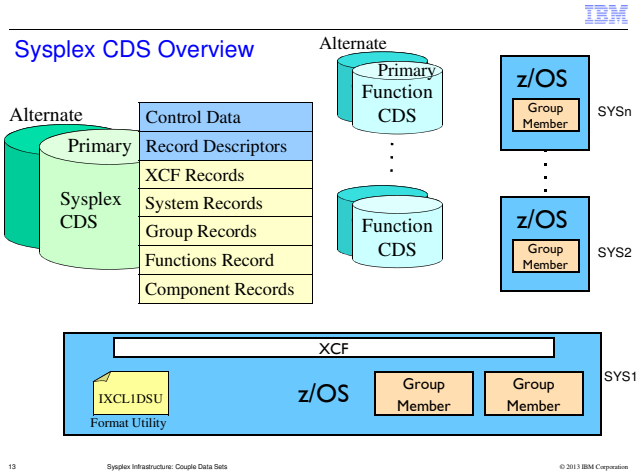
Checkpoint

- At this point, we have our context and sense of topics to be covered
- We are going to look at each type of CDS in turn
  - Usage
  - Content
  - Creation
- We start with Sysplex Couple Data Set
  - Required for every sysplex (except XCF-Local Mode)
  - Our foundation since many concepts equally applicable to other CDS

So we have completed the high level overview that describes what we need to know. It also gives us the context for moving ahead into a more detailed discussion.

To create any couple data set, you need to understand its purpose and in particular, you need to understand how it will be used in a given sysplex. For example, before you create the CFRM CDS, you will need to determine the number of coupling facilities and the number of CF structures in your sysplex. Why? Because the CDS needs to be formatted with enough space to define and manage those resources. So for each CDS we want to understand how it used and what kind of information it holds. Then we can look at the format utility parameters used to create the CDS.

Since every non-trivial sysplex needs a sysplex couple data set, we start there. This discussion will serve as our basis for the function CDS as well. The principles are pretty much the same for them all.



The slide depicts the pertinent sysplex context for the sysplex couple data set (CDS).

The sysplex consists of one or more systems, SYS1 to SYSn running the z/OS operating system. Among other things, the XCF component of z/OS manages all accesses to the couple data sets. On one or more systems, various applications and subsystems have created one or more members of one or more XCF groups. Zero or more functions may be in use by the sysplex. There is a function CDS for each such function. For each CDS, we depict best practices by running with both a primary CDS and an alternate CDS so as to avoid a single point of failure.

The sysplex CDS contains information about every system in the sysplex, every XCF group and member, and every function that is in use by the sysplex.

We depict the format utility IXCL1DSU which is used to create the various couple data sets. We have a chicken and egg problem. Namely, the sysplex CDS can only be defined to a system at IPL time. Thus if you are going to create your first non-trivial sysplex, you will need to IPL a system in XCF-Local mode to run the format utility to create the sysplex CDS. Create a COUPLExx parmlib member that points to the CDS. Then reIPL the system with that COUPLExx parmlib member to start the multi-system sysplex (or monoplex).

Sysplex

- Cluster of z/OS images
- Foundation for high availability, resiliency, horizontal scaling
- In a sense, the sysplex CDS "is" the sysplex
  - Central repository for determining who is and is not in the sysplex

See http://en.wikipedia.org/wiki/Computer_cluster for a brief description of cluster computing in general. All clusters need to solve a similar set of problems:

• Node Management – what nodes are part of the cluster

• Communication – sending messages between nodes

• Data Sharing – accessing and updating shared data with integrity

• Workload Management – which nodes process what work

Clustered computing can be used for many different purposes. The intended purpose and desired attributes of the cluster determines the various design tradeoffs made to solve these problems. The sysplex, which is a cluster of z/OS images, is intended to address business computing needs. The sysplex offers:

• Incremental, granular growth with near linear scalability

• Near continuous availability for both planned and unplanned outages

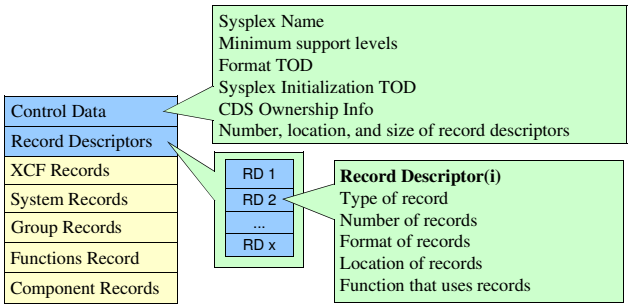
• Automatic, dynamic workload balancing across systems

• High-performance, scalable, read/write access to shared data from all systems in the sysplex -- with integrity

• Ability to incorporate multiple hardware technology families with a wide range of price points, performance, and capacity

Many clusters are built on a distributed model, which generally entails the partitioning of data among the nodes. Sysplex uses a centralized model in which data is equally accessible from every node. So we see a Sysplex CDS used as a centralized repository for node management and a coupling facility to provide high-speed access to shared data. One advantage of the centralized data model vs the partitioned data model is that data remains accessible to the surviving nodes after a node failure. In a partitioned data model, the data managed by the node remains unavailable until the the node is restored to service. A challenge for the centralized data model is the overhead of serializing the data accesses from all the nodes, which explains why there is much concern with the performance of CF lock requests in the sysplex.

Sysplex CDS Content … Controls Common to All CDS



Note: The various content slides highlight just some of the CDS data. These logical representations do not necessarily match the actual physical format in the CDS.

All couple data sets have similar controls to manage the CDS and its content. We describe the controls here once for the Sysplex CDS since they are nearly identical for each type of CDS. However, not every CDS will necessarily use all of the controls indicated.

The **sysplex name** identifies the sysplex that is to use the CDS. To preserve the integrity of the data in the CDS, it can only be used by one sysplex. The name provides a protection mechanism intended to help ensure that a CDS is not accidentally used by the wrong sysplex. The name is set when the CDS is created by the format utility IXCL1DSU. When a CDS is brought into service, XCF rejects the CDS if its sysplex name does not match the name of the acquiring sysplex. If you have different sysplexes with the same sysplex name, you defeat this protection.

When new types of records are added to a CDS, the format utility IXCL1DSU must be updated to create them. The format utility sets the **minimum support leve**l required to make use of the CDS. As a CDS is brought into service, each system inspects the minimum support level to determine whether it has sufficient functionality to use the CDS. During migration, a system might tolerate a CDS with new records but not actually exploit them. A system that can exploit new records might not do so until all systems in the sysplex are at an appropriate level. In some cases, the exploiters of the CDS may have additional data and/or techniques for managing compatibility of CDS usage.

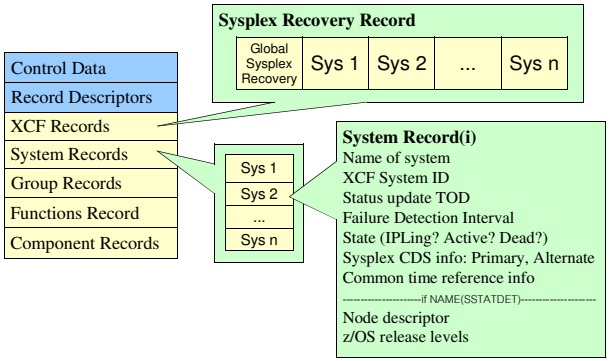
The **format TOD** is a timestamp indicating when the CDS was created by IXCL1DSU. The **sysplex initialization TOD** is a timestamp taken when the first system IPLs into the sysplex. The format TOD is unique to the CDS while the sysplex initialization TOD is unique to the sysplex. If you PSWITCH to a new sysplex CDS, the format TOD would change but the sysplex TOD would persist.

Sysplex ownership information is written to the CDS to “claim” it for use by the sysplex. When a CDS is brought into service, the ownership data is inspected. If the CDS appears to be claimed by another instance of the sysplex, the operator may be prompted to indicate which sysplex should be using the CDS. As applicable, the ownership information would be updated if the prompting sysplex needed to claim the CDS. If so claimed, any other sysplex making use of the CDS loses access to the CDS. Tearing a CDS away from a live sysplex is generally a bad thing to do.

Records in the CDS are identified by names meaningful to the exploiting function. The format utility constructs **record descriptors** used to locate and manage these records within the CDS.

Note: The slides depicting CDS content do not necessarily depict the actual layout of the records. Nor do they necessarily describe all the content.

Sysplex CDS Content … Systems in Sysplex



The Sysplex CDS contains records used to identify and manage systems in the sysplex.

The **Sysplex Recovery Record** is used to manage sysplex-wide processes. For example, this record is used when removing a system from the sysplex. The record indicates which system is being removed. It also indicates which system is managing the removal.

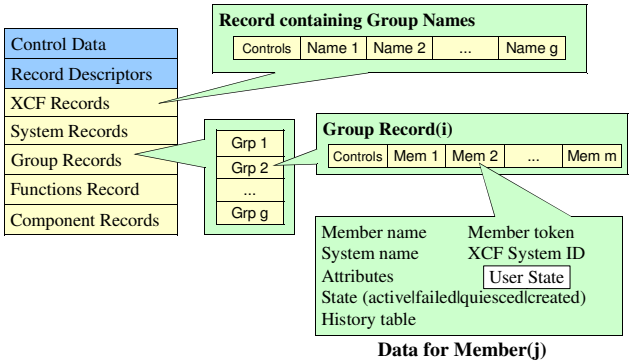
Each system in the sysplex has a **System Record**. The system record contains identifying information (such as the system name, XCF System ID, and hardware node descriptor), status information (such as the status update TOD and system state), system specific parameters (such as its failure detection interval), and configuration data (such as the sysplex CDS in use by the system and information to identify the common time reference being used).

To join the sysplex, an IPLing system must claim a particular system slot for use. The number of slots (system records) is determined by the format utility when the sysplex CDS is created. Generally the IPLing system will reclaim the slot that it occupied when it was last in the sysplex. While trying to join the sysplex, the system is said to be in an **IPLing state**. A system that makes it into the sysplex is said to be in the **active state**. To become active in the sysplex, the system must have access to the same sysplex infrastructure as all the other systems in the sysplex. That is, the system must establish full XCF signal connectivity with every other active system in the sysplex, it must be using the same Sysplex CDS, and it must have access to the same common time reference.

Strictly speaking, the IPLing system does not need access to the same function CDS as the rest of the sysplex in order to become active in the sysplex. However, as a practical matter, it likely will. For example, if a system cannot access the CFRM function CDS it cannot connect to CF structures. A system that is to run in GRS-Star mode will wait-state if it cannot connect to the GRS lock structure during IPL.

Formatting the Sysplex CDS with NAME(SSTATDET) increases the size of the system records, which provides space for the hardware node information needed to allow XCF to use BCPii services to automatically detect when a system has failed (wait-stated). Changing the size of a record create compatibility issues. Down-level systems would neither know what information to put in the new fields nor what information to preserve or clear on behalf of up-level systems. So this parameter is an example of a case for which the minimum support level is helpful (see prior slide).

Sysplex CDS Content … Groups and Members



The Sysplex CDS contains records used to identify and manage the exploiters of XCF in the sysplex. Such exploiters join an XCF group. Each join request creates a member of the group. Thus the various instances of a sysplex application will have one or more members of one or more XCF groups. The capacity of the sysplex CDS with respect to groups and members is determined by installation parameters provided to the format utility IXCL1DSU when the CDS is created.

The Sysplex CDS contains a record that lists the names of the XCF groups.

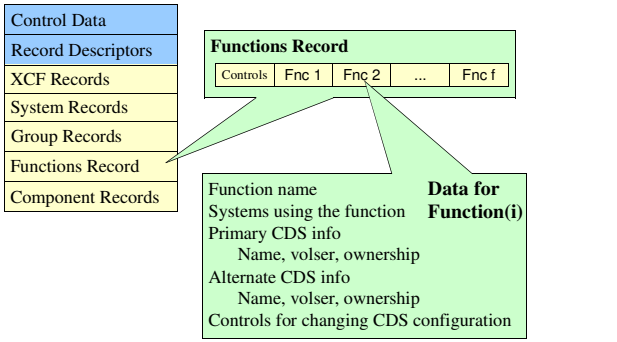
It all contains a group record for each group. The group record lists the members of the group. For each member, the group record contains identifying information (such as member name and member token), status information (such as member state and the system where it resides), member specific attributes, and a small history of significant events (such as member joining the group or updating its user state).

Members can be in one of several **member states**:
Active – member exists, resides on some specific system. Can send signals.
Failed – active member was terminated by XCF due to task failure or system failure
Quiesced – active member asked to be placed in quiesced state.
Created – member exists, but is not associated with any system.
Not Defined – member does not exist

Members with the LASTING=YES attribute (see IXCJOIN macro in the *z/OS Sysplex Services Reference*) can maintain 32 bytes of **user state** information in the Sysplex CDS. This content and interpretation of the user state information is determined by the application. It can be changed dynamically, though applications are discouraged from doing so frequently since it requires an I/O operation to accomplish the update. The user state is visible to and can be updated by other members of the group.

Some applications exploit the member state and/or user state information to control various aspects of member cleanup processing. Use of such data can lead to confusion if the sysplex is IPLed with a copy of the sysplex CDS that contains stale user state information that does not match the state of the application when the sysplex was last active.

Sysplex CDS Content … Functions Information



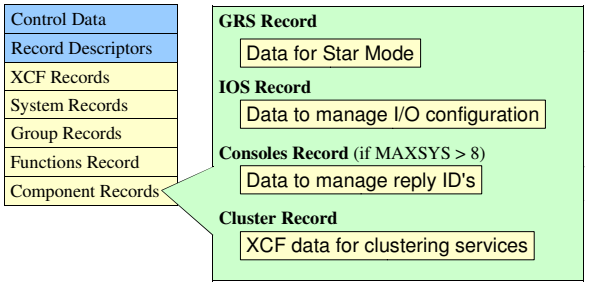
These “functions” are not to be confused with XCF function switches that are enabled or disabled via the FUNCTIONS statement in the COUPLExx parmlib member or the SETXCF FUNCTIONS command.

The Sysplex CDS contains information about the various “functions” in use by the sysplex: CFRM, ARM, SFM, WLM, LOGR, BPXMCDS. Each function makes use of a function couple data set. The Sysplex CDS records information about the primary and alternate function CDS that is in use for each function. Every system in the sysplex that makes use of a given function must be using the same set of function CDS.

A function CDS can only be used by one sysplex at a time. When a function CDS is brought into service, the Control Data Records in the function CDS are written to indicate “ownership” by the sysplex. This ownership information is also recorded in the sysplex CDS. The ownership information provides a protection mechanism intended to help ensure that the function CDS is used only by the owning sysplex. Whenever an I/O operation is initiated against the function CDS, the channel program will include checks to compare the ownership information in the function CDS to the expected ownership information recorded in the sysplex CDS. If the ownership information does not match, the CDS is no longer in use by this sysplex. The I/O operation fails and the CDS is removed from service.

The sysplex CDS also has control information used to manage the sysplex-wide processes that change the set of CDS in use by a function. This information is used when processing a PCOUPLE, ACOUPLE, or PSWITCH command, and when removing from service a CDS that is deemed to have failed.

Sysplex CDS Content … Miscellaneous Data



Finally, the Sysplex CDS has some miscellaneous records.

The **GRS Record** contains information required by z/OS global resource serialization when running in GRS-Star mode. This record exists if DATA TYPE(GRS) NUMBER(1) is specified as an input when the format utility IXCL1DSU creates the Sysplex CDS.

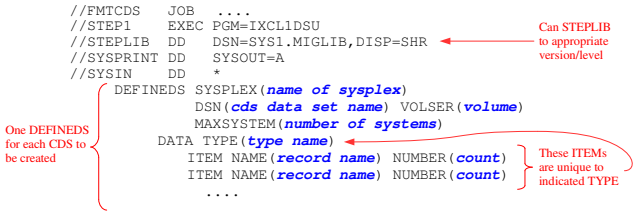
The **IOS Record** contains information required by z/OS I/O Services component to manage changes to the I/O configuration. This information is used by the z/OS Auto-Discovery and Configuration support (zDAC). It is also used when issuing message IOS431I to identify a system holding a RESERVE for a shared device. There is no format utility input related to this record. It always exists.

The **Consoles Record** is created if the Sysplex CDS is formatted to support more than eight systems to help manage reply ID's. When the sysplex has at most eight systems, Consoles manages reply ID's with user state information from one of the members in its XCF group. As the number of systems in the sysplex increases, contention on the Sysplex CDS can arise since the user state information is updated each time a reply ID is assigned. For a large sysplex, systems use the Consoles Record to claim ranges of reply ID's all at once.

The **Cluster Record** contains data required for clustering services. This information describes the sysplex. This record exists if DATA TYPE(CLUSER) NUMBER(1) is specified as an input when the format utility IXCL1DSU creates the Sysplex CDS.

Couple Data Set Format Utility … Overview

- The format utility program IXCL1DSU creates and formats a Couple Data Set (CDS) using parameters that you specify
- These parameters determine type and number of records that need to be created in the CDS



The name of the XCF couple data set format utility is IXCL1DSU. This program resides in SYS1.MIGLIB (which is logically appended to the LINKLIST). The utility is available through STEPLIB, which allows you to run older versions of the utility if you want (for example, from previous z/OS releases).

IXCL1DSU allows you to format all types of couple data sets for your sysplex. The utility contains two levels of format control statements. The primary format control statement, DEFINEDS, identifies the couple data set being formatted. The secondary format control statement, DATA TYPE, identifies the type of data to be supported in the couple data set — Sysplex Couple Data Set (SYSPLEX), Automatic Restart Management (ARM) data, Coupling Facility Resource Management (CFRM) data, Sysplex Failure Management (SFM) data, Workload Manager (WLM) data, z/OS System Logger (LOGR) data, or z/OS UNIX System Services (BPXMCDS) data.

For each DATA TYPE, you specify ITEM statements to identify the type and quantity of data to be supported by the relevant function. The particular ITEM statements that apply to any given function are documented in *z/OS MVS Settting Up a Sysplex (SA22-7625).* You will need to determine whether a given function (DATA TYPE) is needed for a given sysplex, and if so, what type of data records are needed (ITEM NAME) and how many (NUMBER).

Couple Data Set Format Utility … SYSPLEX CDS

```
DEFINEDS
     …
     MAXSYSTEM(#systems in sysplex)
DATA TYPE(SYSPLEX)
     ITEM NAME(GROUP) NUMBER(#groups to support)
     ITEM NAME(MEMBER) NUMBER(#members per group)
     ITEM NAME(GRS) NUMBER(1)          <-- If GRS-Star mode
     ITEM NAME(CLUSTER) NUMBER(1)
     ITEM NAME(SSTATDET) NUMBER(1)     <-- A "must have"
```

Control Data
Record Descriptors
XCF Records
System Records
Group Records
Functions Record
Component Records

- Note more kinds of records than ITEMS
- Some choices are release dependent
  - ITEMs, even NUMBER values
  - Implies CDS "versioning"
- Choices might impact number of records or their size, and protocols
- Discovering groups and members ?
- Allow for growth, but ....

See SYS1.SAMPLIB(IXCSYSPF)

Personally, I would make MAXSYSTEM be the exact number of systems in the target sysplex. Traditionally, it is suggested that you allow some white space so that the number of systems in the sysplex can easily grow. Doing so saves you the trouble of having to later format and bring into service the larger CDSes when the new system finally joins the sysplex. You may want to adjust the "white space" parameters on the XCF_SYSPLEX_CDS_CAPACITY health check accordingly.

Discovering the number of groups and number of members per group takes the most research. The idea is to choose values large enough to support the various system components and applications that will be running in the sysplex, but not so large as to unnecessarily impact I/O performance (see next slide). If you already have a sysplex up and running, issue the D XCF,COUPLE command to determine the peak number of groups and members ever defined in the Sysplex CDS. These peaks could provide guidance as to reasonable values – under the assumption that the past workload will be similar to the future workload (with respect to XCF groups and members). You might also be able to extrapolate reasonable values from a test sysplex whose application set is sufficiently close to your production sysplex.

If you are creating a new sysplex, you might (erroneously) think to (1) Format the Sysplex CDS with really large values to ensure that the various applications can create their groups and members; (2) Bring up the sysplex; (3) Use D XCF,COUPLE to see what values are actually needed for the sysplex; and then (4) Create a Sysplex CDS of a more appropriate (smaller) size. However, this technique is not practical in general as it requires a sysplex IPL to bring a smaller Sysplex CDS into service.

Thus we must discover in advance what will be needed. *Setting Up A Sysplex* describes the groups used by various z/OS components. Other XCF exploiters should document their group and member needs so that you can do appropriate planning. The challenge is to determine the complete set of components, subsystems, and applications that will exploit XCF in your new sysplex and then find the information describing their needs.

In any case, I suggest that you maintain a list of groups and the associated applications. This list will be helpful not only for future CDS sizing exercises, but will likely prove invaluable when trying to diagnose sysplex problems.

Specify other ITEMs as needed. Not every record in the Sysplex CDS has a corresponding ITEM. Some choices for NAME or NUMBER are restricted to use by certain z/OS releases or maintenance levels. If so, you must upgrade to the appropriate software throughout the sysplex before bringing into service a CDS formatted with such values. Use of a CDS by systems that do not understand the new ITEMs or don't support the increased NUMBER can lead to failures. Thus there are various techniques used to "version" the CDS. In general, the presence of a down-level system in the sysplex prevents an up-level CDS from being brought into service. If an up-level CDS is in use by the sysplex, a down-level system cannot join the sysplex. A sysplex-wide IPL is needed to revert to a back-level CDS.

SSTATDET is required for XCF to exploit BCPii services to automatically detect and remove failed systems. The parameter increases the size of the system records to allow for additional system related data. Even if XCF cannot exploit BCPii in your sysplex, specify SSTATDET. The expanded system records have other useful data.

Perils of too much white space

Charts extracted from IBM Redbook: SG24-7816
"System z Mean Time To Recovery Best Practices"

- White space implies
  - Longer records
  - More records
- Which can elongate
  - CDS I/O response time
  - IPL time
  - ACOUPLE time
  - PSWITCH time
- Can dynamically grow as needed, but changing to a smaller CDS requires:
  - Sysplex IPL
    - For Sysplex or CFRM CDS
  - Inducing function failures
    - For ARM, SFM, WLM, LOGR, or BPXMCDS CDS

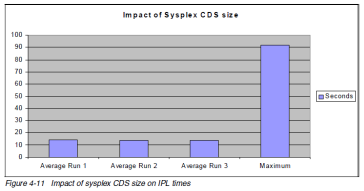


Figure 4-11 Impact of sysplex CDS size on IPL times

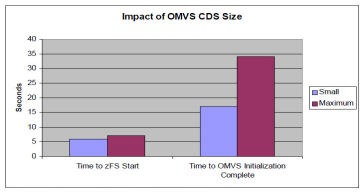


Figure 5-2 Results of BPXMCDS size on OMVS initialization time

Planning for reasonable expansion does not mean specifying the maximium possible size definitions for the CDSes. The larger the CDS, the more processing it requires. If the records are all empty, reading them all in induces unnecessary overhead. If a drastically oversized SYSPLEX CDS become the primary SYSPLEX CDS, the only way to reduce the size is via a sysplex IPL pointing at newly formatted smaller SYSPLEX CDSes. The same is true for the CFRM CDS, the only way to reduce the size is via sysplex IPL. Rather than drastically oversize, allocate SYSPLEX CDSes and CFRM CDSes that allow for growth and if necessary define and activate larger SYSPLEX CDSes or CFRM CDSes as needed in the future.

Similarly, SFM, ARM, LOGR, WLM and BPXMCDS should not be drastically oversized. A sysplex IPL is not required to bring in smaller CDSes for this set. However, a PSWITCH must be done so that only the primary CDS is in-use. Then the device on which the primary CDS resides must be forced offline. The function exploiting the CDS and the services it provides may encounter significant failures during this activity.

Creating Sysplex CDS Content

- Nothing for you to do beyond:
  - Formatting the CDS
  - Bringing CDS into service
- All data is updated dynamically by the systems in the sysplex

The actual content of the sysplex CDS is dynamically created by the systems in the sysplex. The installation does not directly store any data in the CDS. Thus there is nothing for the installation to do beyond using the format utility IXCL1DSU to create the data set and bringing the CDS into service (generally through specifications in the COUPLExx parmlib member, or possibly via the SETXCF COUPLE command).

In contrast, most functions require the installation to create data (policies) in their function CDS. A function CDS will typically be a mixture of this static installation defined data as well as transient data created dynamically by the systems in the sysplex.

Function Couple Data Sets

- CFRM Coupling Facility Resource Manager
- ARM Automatic Restart Manager
- SFM Sysplex Failure Manager
- WLM Workload Manager
- LOGR z/OS Logger
- BPXCMDS z/OS UNIX System Services

Let's look at each in turn ....
- Usage
- Content
- Creation

Having completed our overview of the Sysplex CDS, we now want to similarly examine each of the function couple data sets. Our objective is to be able to use the format utility IXCL1DSU to create the function CDS. This is done in much the same way as it was done for the sysplex CDS. However, each type of CDS will have its own unique input parameter ITEMs. To format the CDS we need to understand these ITEMs. To understand the ITEMs, we need to understand the intended purpose of the function and the type of data that must be maintained in the function CDS in order for the function to function.

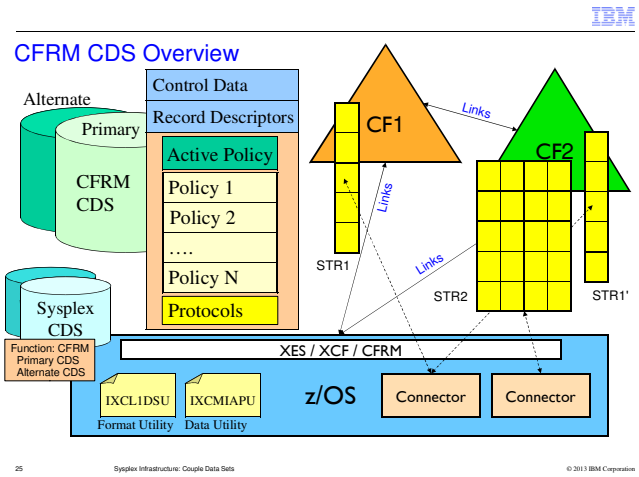
The handout includes detailed slides for each type of CDS, but due to time constraints we cannot cover them all during the presentation. However, most of the function CDS have broadly similar concepts. My goal for the presentation will be to outline the broad concepts and highlight the key differences.

The CFRM, ARM, and SFM function CDS are nearly identical in terms of how they are created and managed, differing only in the specific ITEMs used when formatting the CDS. So I'll present the CFRM function CDS in detail and leave the details for ARM and SFM to the handout.

The WLM function CDS has similar concepts to CFRM, but differs in the mechanics of how the content of the CDS is created. So the presentation will focus on those differences.

When discussing the LOGR function CDS, we use many of the same terms as are used when discussing the CFRM function CDS. However, there is fundamentally something different going on with LOGR. The presentation will explain.

Finally, the BPXMCDS function CDS is more similar to the sysplex CDS than to the other function CDS. The presentation will explain.



The slide depicts the pertinent sysplex context for the Coupling Facility Resource Manager (CFRM) function couple data set (CDS).

The sysplex consists of one or more systems, SYS1 to SYSn running the z/OS operating system (though the diagram only shows one). Among other things, the XCF component of z/OS manages all accesses to the various CDS. On one or more systems, various applications and subsystems have created one or more connections to one or more coupling facility (CF) structures. In the diagram, we see a connections (represented by dashed lines) to two different structures, STR1 and STR2. Structure STR1 is a duplexed structure with two structure instances, one in each of coupling facility CF1 and CF2. The XES component uses the CFRM CDS to manage the coupling facilities, the structures residing therein, and the connectors to those structures. The CECs on which the z/OS images reside are connected to the coupling facilities via CF links, as are the two CFs themselves (required for system managed structure duplexing).

The sysplex CDS contains information about the CFRM function, and in particular, the primary and alternate CFRM CDS. The CFRM CDS contains installation specified administrative policies. The policies describe how the coupling facility resources are to be managed. Different policies might be used to achieve various objectives based on business needs. The installation chooses one of the policies to be the active policy. The active policy contains a point in time copy of the administrative policy, plus status information to describe the current state of the CF related resources. For both CDS, we depict best practices by running with both a primary CDS and an alternate CDS so as to avoid a single point of failure.

We depict the format utility IXCL1DSU which is used to create the couple data sets. We depict the administrative data utility IXCMIAPU which is used to create policies in the CFRM CDS that define how the CF resources are to be managed.

CFRM Policies Are Used to Manage CF Resources

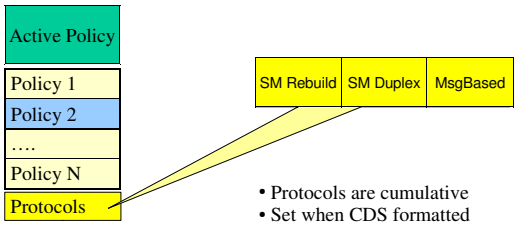
- Define coupling facilities to be used by the sysplex
  - CF can only be used by one sysplex at any one time
- Define structures that can be used by the sysplex
- For each structure you specify:
  - Name
  - List of CFs that are candidates to host structure
  - List of other structures that should not be in the same CF as this one
  - Amount of space structure can consume in host CF
  - Protocols and thresholds to be applied to the structure
    - Duplexing of structure content in an alternate CF
    - Dynamic reconfiguring of structure to match exploiter usage
    - Full threshold for alerts and reconfiguration actions
    - Connectivity threshold at which to initiate rebuild of structure

To set up your CFRM policy, you must know which subsystems and applications in your installation are making use of the coupling facility and what their structure requirements are. Each application specifies the structure types required and provides information as to the name and size of the structure or structures it uses. Based on this input, you must determine an appropriate number of coupling facilities and how best to distribute the structures within the coupling facilities.

To define a CFRM policy, you must:

- Identify each coupling facility.
- Define the amount of space within the coupling facility that you want for dumping.
- Identify each structure. For each structure you specify:
  - The size of each structure, the maximum size and optionally, an initial size and a minimum size.
  - A percentage value that represents a percent full threshold for the structure.
  - Whether you want the system to automatically alter the size of a structure when it reaches a user-defined or system-defined percent full threshold.
  - Whether the structure is to be duplexed
  - A “preference list” of coupling facilities in which the structure may be allocated.
  - A list of structures, an “exclusion list”, that are not to be allocated in the same coupling facility as the structure.
  - A percentage value to indicate the amount of lost connectivity to a structure that the sysplex can tolerate before z/OS initiates a structure rebuild, if applicable.

CFRM CDS Content – CFRM Protocols



The Coupling Facility Resource Manager CFRM) couple data set (CDS) contains three different types of information: the active policy, one or more administrative policies, and protocols. This slide depicts the protocols.

The protocols supported by the CFRM CDS are determined by the input parameters to the format utility IXCL1DSU used to create the CFRM CDS:

ITEM NAME(SMREBLD) NUMBER(1)
ITEM NAME(SMDUPLEX) NUMBER(1)
ITEM NAME(MSGBASED NUMBER(1)

SMREBLD implies system managed rebuild processing is supported.
SMDUPLEX implies system managed CF structure duplexing is supported.
MSGBASED implies CFRM message based processing is supported.

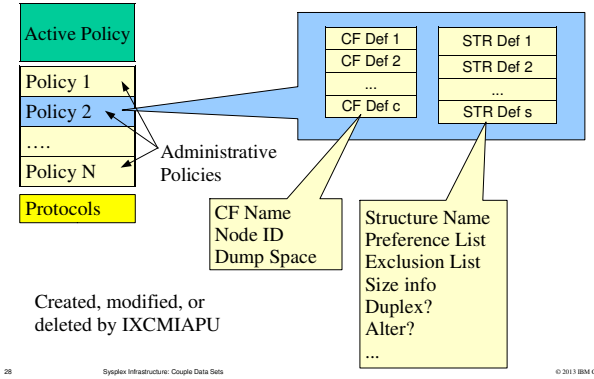
Note that the protocols are cumulative so you need only specify the highest desired protocol (MSGBASED implies SMDUPLEX and SMREBLD are supported, SMDUPLEX implies SMREBLD is supported). Everyone should be using MSGBASED as it provides significant improvement to elapsed recovery time after events such as system failure or loss of connectivity to a coupling facility.

Some installations have reservations regarding use of system managed CF structure duplexing due to the increased service time that may occur for some CF requests issued to a duplexed structure. However, this consideration is not relevant to the format of the CFRM CDS. A CFRM CDS formatted to support SMDUPLEX is a necessary condition for use of system managed CF structure duplexing, but it is not sufficient. A structure will not be duplexed unless the CFRM policy permits it.

MSGBASED processing has already proven the test of time. It can be disabled dynamically if the need should arise. Thus there are no reasonable impediments to formatting the CFRM CDS with MSGBASED.

All systems running with currently supported releases are compatible with any of these protocols.

CFRM CDS Content - Administrative Policies



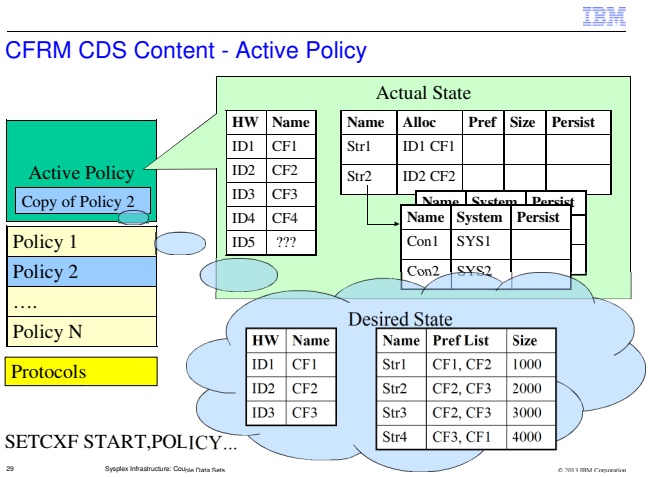
The Coupling Facility Resource Manager CFRM) couple data set (CDS) contains three different types of information: the active policy, one or more administrative policies, and protocols. This slide depicts the administrative policies.

Each administrative policy defines how various coupling facility resources are to be managed. Any given policy defines the particular coupling facilities to be used by the sysplex and the various coupling facility structures to be created in those coupling facilities. For a coupling facility, the policy indicates the Node ID of the particular image on the particular CEC that is to be the CF, the name by which you want the CF to be known, and the amount of dump space to be set aside in the CF for diagnostic dumps of structures. For each structure, the policy indicates the name of the structure, a preference list indicating the set of coupling facilities where the structure can be allocated, an exclusion list indicating the names of structures that should not be located in the same CF as the structure being defined (for failure isolation), information about how much storage the structure can consume in the CF, whether the structure should be duplexed, whether the structure should be dynamically re-sized in response to various conditions and thresholds, and … there are more, but you get the idea.

It is often suggested that you might need different policies to address the need for different business objectives – perhaps one for prime shift operations and another for your batch window. If nothing else, you might want to maintain multiple policies for good change control. When a change is needed, you could define and activate a new policy while maintaining the old policy in case there is a need to fall back. In the past, multiple policies were needed to temporarily remove a CF from the configuration in order to perform disruptive maintenance on the CF. That motivation has fallen by the wayside with the appearance of support for “maintenance mode” and REALLOCATE.

The content of these administrative policies is defined by the administrative data utility IXCMIAPU which is found in SYS1.MIGLIB. IXCMIAPU is sometimes called the administrative policy utility. IXCMIAPU can be used to create, delete, update, or list an administrative policy in the CFRM CDS. Input to IXCMIAPU indicates the name of the policy to be manipulated. If creating or updating a policy, the inputs define the various coupling facilities and structures as described above.

In order to create the CFRM CDS with the format utility, you will need to know how many policies you want to define, the maximum number of coupling facilities to be defined in any one policy, and the maximum number of structures to be defined in any one policy. This information is needed when using the format utility IXCL1DSU to create a CFRM CDS.

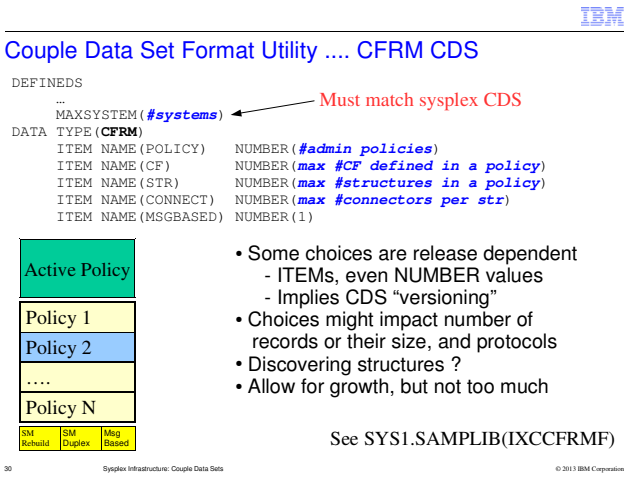


The Coupling Facility Resource Manager CFRM) couple data set (CDS) contains three different types of information: the active policy, one or more administrative policies, and protocols. This slide depicts the active policy.

An operator command is used to activate (start using) a particular policy:
SETXCF START,POLICY,TYPE=CFRM,POLICY=policyname

When activated, the current copy of the named administrative policy is copied into the active policy record in the CFRM CDS (depicted as the blue cloud “desired state”). In the diagram, we have activated “policy 2”. Let me clearly emphasize, this is a point in time copy taken when the policy is started. When you use the administrative data utility IXCMIAPU to change a policy, you update the administrative copy of the policy. Changing the administrative copy of “policy 2” does not cause those changes to be reflected in the active copy of “policy 2”. If you want the revised “policy 2” to be used by the sysplex, you must start the policy again to get the new “policy 2” copied to the active policy. So do not be confused. Even though the active policy and the administrative policy might both be called “policy 2”, they are really two different entities.

In the diagram the green box “actual state” depicts the fact that the active policy also contains information describing the current state of the coupling facility related resources. There is information about the coupling facilities in use, the structures, and the structure connectors. This information is updated dynamically by the systems in the sysplex. Note that the active policy may be managing CF's and structures that are not defined in the administrative policy that was started. In the diagram, CF4 is not defined in “policy 2”. The presence of CF4 in the active copy of the policy implies that CF4 was defined in some previously started policy and is still in use by the sysplex. Since CF4 is not defined in “policy 2” it will be deleted when the sysplex no longer has need of it, which is to say, when it no longer contains any structures. Though not shown, structures still in use from a prior policy but no longer defined in the current policy would similarly persist in the active policy until they were deleted. Structures and CFs in this state are said to be “pending delete”. Also note that the active policy shows the existence of CF5. In this case, the CF was not defined in any policy (and so has no name), but the sysplex is aware of the CF because it is physically connected.



To format the CFRM CDS, you need to determine the desired number of administrative policies, the maximum number of coupling facilities to be defined in any one policy, the maximum number of structures to be defined in any one policy, and the maximum number of connections to be established to any one CF structure. You probably want to allow some white space for creating an additional policy if the need should arise. Discovering the number of structures needed will likely require the most research as you will need to determine the complete list of sysplex applications that will be configured to exploit CF structures. The number of connections would likely need to be greater than or equal to the maximum number of systems in the sysplex since most applications will have at least one connection from each system in the sysplex. Some applications might have more than one connection per system. Again, this requires research.

The MAXSYSTEM specified for the CFRM CDS should equal the MAXSYSTEM specified for the sysplex CDS. This requirement is not enforced, though the health check XCF_CDS_MAXSYSTEM will complain if the MAXSYSTEM value for the CFRM CDS is smaller than the MAXSYSTEM value for the primary sysplex CDS. If the MAXSYSTEM for CFRM CDS exceeds that of the sysplex CDS, the CFRM CDS will be unnecessarily big which can induce I/O delays and related performance issues. If smaller, a system in the sysplex might not be able to use the CFRM CDS (which could cause a wait-state during IPL).

The following statements are not specified because they are implied by MSGBASED:
-ITEM NAME(SMREBLD) NUMBER(1)
-ITEM NAME(SMDUPLEX) NUMBER(1)
Everyone should be running with CFRM making use of MSGBASED protocols, which helps reduce the amount of contention on the CFRM CDS. The time to accomplish recovery actions such as structure rebuild and duplex fail over is much less with message based protocols as opposed to policy based protocols. It should not be a problem, but for completeness be aware that your systems must be running z/OS v1R8 or later in order to IPL into a sysplex that is using a CFRM CDS created with MSGBASED.

In general, use of certain ITEMs or NUMBER values could require some minimum level of z/OS software for the CDS to be used by that system. As of this writing, all supported releases can tolerate any valid specification.

SYS1.SAMPLIB(IXCCFRMF) contains a sample job to format a CFRM CDS.

Creating CFRM CDS Content

- You must
  - Format the CFRM CDS
  - Bring the CDS into service by the sysplex
  - Create one or more administrative policies
  - Choose and activate the policy to be used by the sysplex
- The CFRM CDS contains data from different sources
  - Protocols — Static data set by Format Utility (you)
  - Administrative policies — Static data set by Data Utility (you)
  - Active policy — Dynamic data set by systems in the sysplex
- Your influence on the active policy is limited to the exact point in time that you activate a given static copy of a policy
  - When you activate a policy, the static copy of the administrative policy is copied into the active policy
  - If you later change the static copy, it does nothing to the active copy
  - Sysplex only looks at an administrative policy when you activate it

This slide is intended as a reference. It summarizes how the CFRM CDS comes to have content. We will have a similar slide for each of the different function CDS. We'll use these slides to compare and contrast the various function CDS.

The installation needs to create policies (content) in the CFRM CDS and activate one of them for the CDS to be of any use. First you must use the format utility IXCL1DSU to create the CFRM CDS. You must already have some idea of what your policies will be in order to specify appropriate inputs to ICXL1DSU. After creating the CFRM CDS, use the administrative data utility IXCMIAPU to create one or more administrative policies in the CDS. If the sysplex is not currently using any CFRM CDS, you can create the policies in the CDS before bringing it into service. More typically, IXCMIAPU manipulates the policies in the CFRM CDS that is currently in use by the sysplex. Regardless, the CFRM CDS will at some point be the primary CDS in use by the sysplex and it will contain the administrative policies that you have defined. One of the administrative policies must then be activated for use by the sysplex.

The protocols supported by the CFRM CDS are fixed at format time. The protocols will not be applicable until the CDS becomes the primary CFRM CDS in use by the sysplex. Since you specify the inputs to the format utility, you have control over which protocols apply.

I describe the administrative policies as being static since they are only changed at your direction through use of IXCMIAPU. The systems in the sysplex do not otherwise dynamically modify the content of the administrative policies. Indeed, they are only functionally accessed when a policy is started.

The content of the active policy is dynamically created as systems perform their processing. When you activate a particular policy, a copy of that administrative policy is stored in the active policy. You might argue that this copy is static, but I want you to think of the entire active policy as being totally under control of the systems in the sysplex so as to disabuse you of the notion that the active policy is in any way changed when IXCMIAPU updates the administrative copy of a policy by the same name. In terms of managing the CF resources, the portion of the active policy that retains the current status of the resources is critical to the operation of CFRM. This status information is dynamically created by the systems in the sysplex.

To be clear: when you start a policy, a point in time copy of the administrative policy is stored in the active policy. When you use IXCMIAPU to change a policy, you update the administrative copy, not the active copy. If you want the revised policy to be used by the sysplex, you must start the policy again. Even though the active policy and the administrative policy have the same name, they are really two completely separate entities.

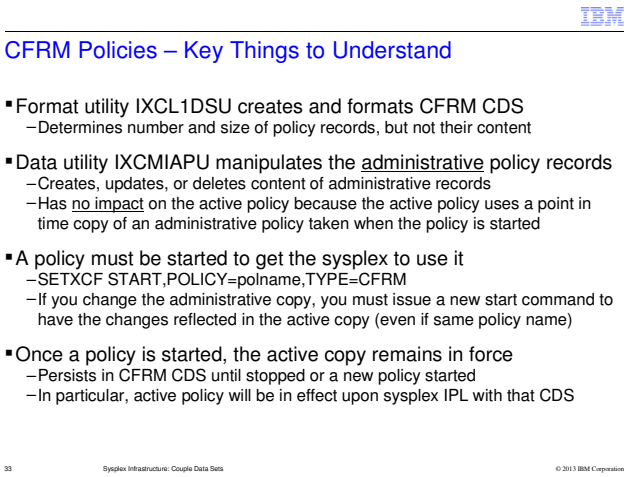
Creating CFRM Policies

- The data utility program IXCMIAPU is used to manipulate administrative policies in a Couple Data Set formatted for CFRM
- The parameters you specify define the policy, which in turn describes how CFRM is to manage coupling facility resources
- The data utility can
  - Create new policies
  - Replace existing policies
  - Delete policies
  - Report on the policies currently defined in the CDS
- See SYS1.SAMPLIB(IXCCFRMP)

This slide is intended as a reference. It summarizes how CFRM policies are created. We will have a similar slide for each of the different function CDS. We'll use these slides to compare and contrast the various function CDS.

The policies and specifics of creating them are beyond the scope of this presentation. However, I hope you now have enough context to be able to understand the material in *z/OS MVS Setting Up A Sysplex* that describes how to use administrative data utility IXCMIAPU to create the polices that would be appropriate for your installation.

SYS1.SAMPLIB(IXCCFRMP) has a sample job that creates a policy in a CFRM CDS.

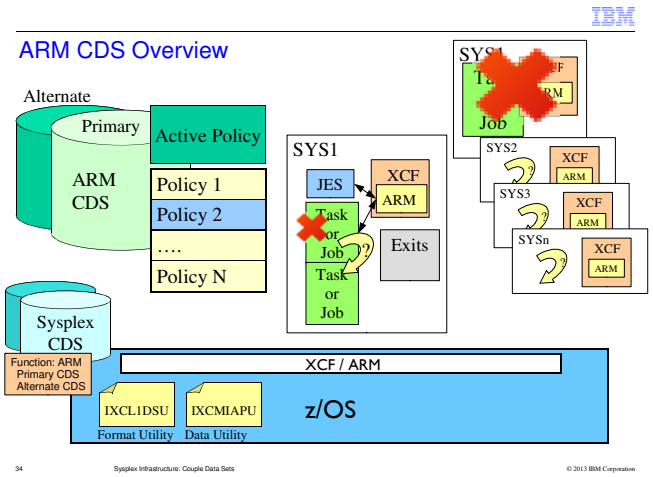


This slide is intended to summarize the key concepts about the CFRM CDS. These concepts directly apply to the ARM CDS and the SFM CDS since they also have "administrative policies" and an "active policy".

I know I've repeatedly emphasized this point: updates to the administrative policy do not impact the active policy, and you must start the updated policy in order for the changes to be applied to the active policy. I do so because it appears that people continue to be tripped up by the failure to understand this behavior.

Suppose the sysplex is making use of a CFRM CDS and you start a policy POL1. If you should later happen to reIPL the sysplex with that same CFRM CDS, policy POL1 will still be active. In most cases this is exactly what you would want. The active policy will still contain state information about the CF resources that were in use before the sysplex went down. As the sysplex comes up, the active policy will be updated to indicate the demise of the connectors from the prior sysplex instance: some will be deleted outright, others marked as failed. Depending on various persistence attributes and the state of the relevant CF, some structures might be deleted while others will persist. As the workload restarts, applications may take recovery actions based on various factors such as the existence of failed connectors, the content of their structures (if they survived), and any data that might have been hardened to DASD prior to the sysplex failure (such as log records).

If you reIPL the sysplex with freshly formatted CFRM CDS, there will not be an active policy. If you need a policy to be active when the first system IPLs (prior to the point where a SETXCF START,POLICY command can be entered), you will need to specify the CFRMPOL keyword in the COUPLExx parmlib member of the first system to IPL into the sysplex. The CFRMPOL keyword indicates the name of the CFRM policy that is to be started at IPL time if there is no active policy in the CFRM CDS. This issue tends to come up for installations running GRS-Star mode. GRS wait-states an IPLing system if it cannot get to the ISGLOCK structure (as would be the case if there is no active CFRM policy).



The slide depicts the pertinent sysplex context for the Automatic Restart Manager (ARM) function couple data set (CDS).

The sysplex consists of one or more systems, SYS1 to SYSn running the z/OS operating system. Among other things, the XCF component of z/OS manages all accesses to the various CDS. On one or more systems, various applications and subsystems have registered with ARM. In the diagram, we see two examples of what ARM might do:

•In the center, we see that a task or job on system SYS1 has failed. The failed task or job had previously registered with ARM, so ARM took notice when it failed. ARM examined the active policy in the ARM CDS and in conjunction with various exit routines, determined how to proceed. In this case, ARM arranged for the task or job to be restarted on the same system. ARM works in conjunction with JES (as needed) to accomplish the restart.

•To the right, we have an example of a cross-system restart. In this case, the task or job failed because the system on which it was running failed (SYS1). One of the ARM instances on a surviving systems detects the failure of SYS1. Based on information in the active ARM policy and the current state of the sysplex, ARM will selects one of the surviving systems to host the restarted task or job.

The sysplex CDS contains information about the ARM function, and in particular, the primary and alternate ARM CDS. The ARM CDS contains installation specified administrative policies. The policies describe how the ARM is to handle the restarting of failed jobs or tasks. Different policies might be used to achieve various objectives based on business needs. The installation chooses one of the policies to be the active policy. The active policy contains a point in time copy of the administrative policy, plus status information to describe the current state of the jobs or tasks that have registered with ARM. For both CDS, we depict best practices by running with both a primary CDS and an alternate CDS so as to avoid a single point of failure.

We depict the format utility IXCL1DSU which is used to create the couple data sets. We depict the administrative data utility IXCMIAPU which is used to create policies in the ARM CDS that defines ARM is to behave.

ARM Policies Control Restart of Work After a Failure

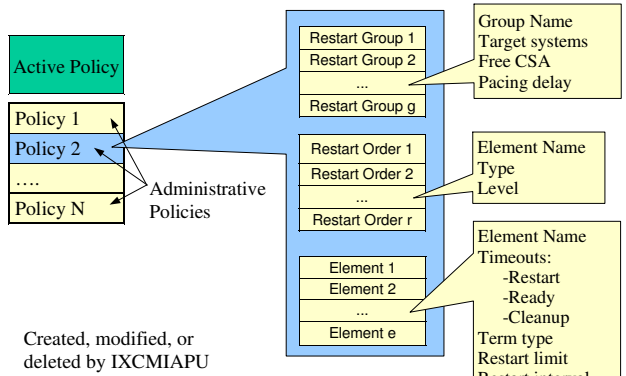
- Whether restarts will occur for some, none, or all elements
- The method by which an element is to be restarted
- Grouping and dependencies among elements
- Which systems are candidates for hosting a restarted element
- Restart intervals, thresholds, and time out values

- Note: the policy that governs the restart is the policy that is active when ARM processes the restart, not the policy that was active when the element registered with ARM

Use the automatic restart management policy to specify how batch jobs and started tasks that are registered as elements of automatic restart management (ARM) should be restarted. The policy can specify different actions to be taken when a system fails, and when an element fails. Automatic restart management uses the IXC_WORK_RESTART exit, the IXC_ELEM_RESTART exit, the event exit, and the IXCARM macro parameters, in conjunction with the automatic restart management policy (the specified values and the defaults) when determining how to restart elements

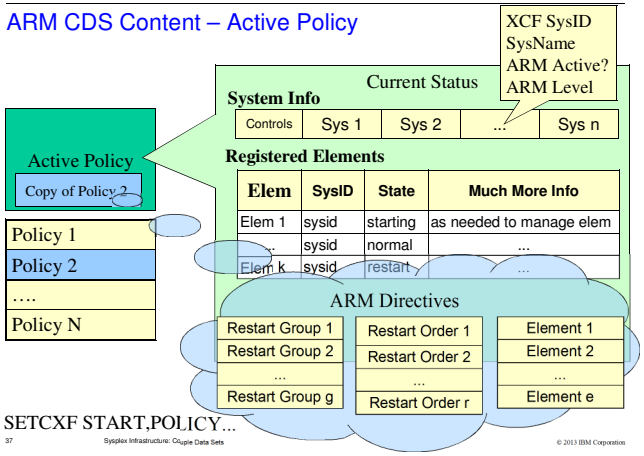
The policy used to restart an element or a group of elements is the one that is active when the element is restarted, not the one that was active when the element registered.

ARM CDS Content – Administrative Policies



Created, modified, or deleted by IXCMIAPU

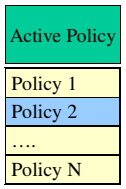
ARM CDS Content – Active Policy



SETCXF START,POLICY...

Couple Data Set Format Utility .... ARM CDS

```
DATA TYPE(ARM)
     ITEM NAME(POLICY) NUMBER(#admin policies)
     ITEM NAME(MAXELEM) NUMBER(max #elements per policy)
     ITEM NAME(TOTELEM) NUMBER(max #registered elements)
```



- MAXELEM determined by your policies
- Number of restart groups and restart elements is a function of MAXELEM
- Determining TOTELEM ?

See SYS1.SAMPLIB(IXCARMF)

The number of records for restart groups and restart order is a function of the number of elements.

Creating ARM CDS Content

- You must
  - Format the ARM CDS
  - Create zero or more administrative policies
    - There is an implicit “no name” default policy
  - Choose and activate the policy to be used by the sysplex
- The ARM CDS contains data from different sources
  - Administrative policies — Static data set by Data Utility (you)
  - Active policy — Dynamic data set by systems in the sysplex
- Your influence on the active policy is limited to the exact point in time that you activate a given static copy of a policy
  - When you activate a policy, the static copy of the administrative policy is copied into the active policy
  - If you later change the static copy, it does nothing to the active copy
  - Sysplex only looks at an administrative policy when you activate it

To start automatic restart management with the policy defaults, issue:

SETXCF START,POLICY,TYPE=ARM

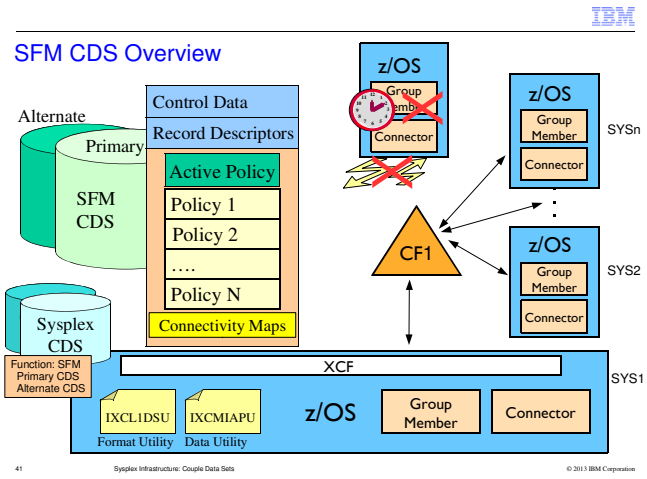
To start your own automatic restart management policy (mypol, for example), issue:

SETXCF START,POLICY,TYPE=ARM,POLNAME=mypol

Use the SETXCF STOP command to disable automatic restarts.

Creating ARM Policies

- The data utility program IXCMIAPU is used to manipulate administrative policies in a formatted ARM Couple Data Set
- The parameters you specify define the policy, which in turn describes how ARM is to behave
- The data utility can
  - Create new policies
  - Replace existing policies
  - Delete policies
  - Report on the policies currently defined in the CDS
- See SYS1.SAMPLIB(IXCARMP0)

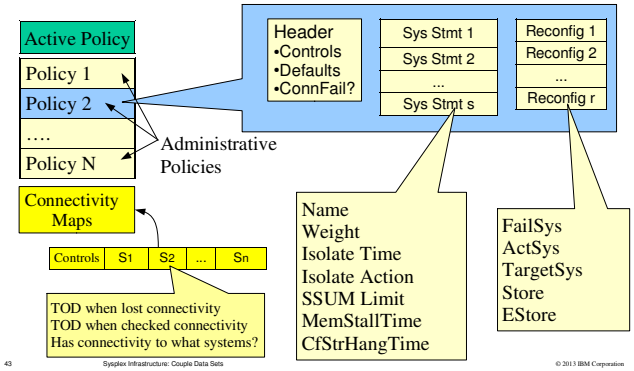


SFM deals with the detection and resolution of sympathy sickness conditions that can arise when a system or a sysplex application is unresponsive.

SFM Policies Help Manage Sympathy Sickness

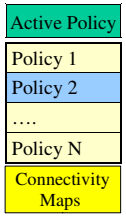
- The systems in the sysplex can detect various problems
  - Inability of a system to participate in a sysplex
  - Sick but not dead issues
- The SFM Policy determines whether and how the sysplex is to resolve these problems
  - Unresponsive system
  - Loss of signal connectivity
  - Unresponsive group member
  - Unresponsive connector

SFM CDS Content



Couple Data Set Format Utility .... SFM CDS

```
DEFINEDS
      …
      MAXSYSTEM(#systems)
DATA  TYPE(SFM)
      ITEM NAME(POLICY)    NUMBER(#admin policies)
      ITEM NAME(SYSTEM)    NUMBER(max #system statements in a policy)
      ITEM NAME(RECONFIG)  NUMBER(max #reconfig statements in a policy)
```



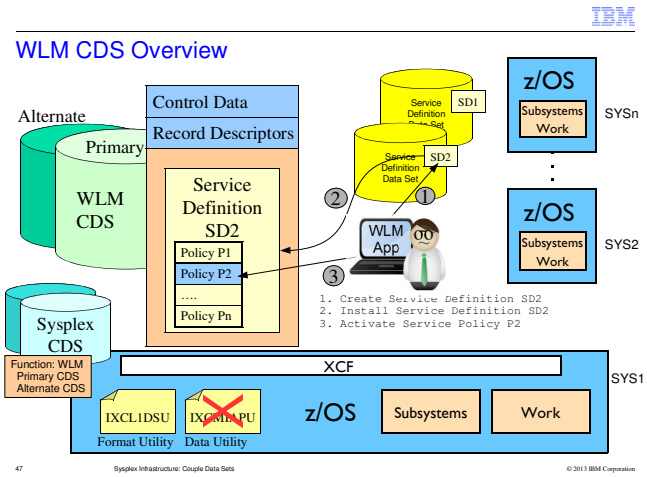
- MAXSYSTEM vs ITEM NAME(SYSTEM)
- Anyone using RECONFIG?

Creating SFM CDS Content

- You must
  - Format the SFM CDS
  - Create one or more administrative policies
  - Choose and activate the policy to be used by the sysplex
- The SFM CDS contains data from different sources
  - Administrative policies — Static data set by Data Utility (you)
  - Active policy — Dynamic data set by systems in the sysplex
- Your influence on the active policy is limited to the exact point in time that you activate a given static copy of a policy
  - When you activate a policy, the static copy of the administrative policy is copied into the active policy
  - If you later change the static copy, it does nothing to the active copy
  - Sysplex only looks at an administrative policy when you activate it

Creating SFM Policies

- The data utility program IXCMIAPU is used to manipulate administrative policies in a formatted SFM Couple Data Set
- The parameters you specify define the policy, which in turn describes how SFM is to behave
- The data utility can
  - Create new policies
  - Replace existing policies
  - Delete policies
  - Report on the policies currently defined in the CDS
- See SYS1.SAMPLIB(IXCSFMP)



MVS workload management provides a solution for managing workload distribution, workload balancing, and distributing resources to competing workloads. MVS workload management is the combined cooperation of various subsystems (CICS®, IMS/ESA®, JES, APPC, TSO/E, z/OS UNIX System Services, DDF, DB2®, SOM, LSFM, and Internet Connection Server) with the MVS workload management (WLM) component.

The service level administrator is responsible for defining the installation's performance goals based on business needs and current performance. This explicit definition of workloads and performance goals is called a service definition. A service definition resides in a service definition data set. An ISPF application is used to manage service definitions. With a service definition, you can specify goals for all MVS managed work, whether it is online transactions or batch jobs. The goals defined in the service definition apply to all work in the sysplex.

Performance management is the process workload management uses to decide how to match resources to work according to performance goals. Workload management algorithms use the service definition information and internal monitoring feedback to check how well they are doing in meeting the goals. The algorithms periodically adjust the allocation of resource as the workload level changes.

For each system, workload management handles the system resources. Workload management coordinates and shares performance information across the sysplex. How well it manages one system is based on how well the other systems are also doing in meeting the goals. If there is contention for resources, workload management makes the appropriate trade-offs based on the importance of the work and how well the goals are being met.

A two-step process is required before the sysplex starts using a new service definition. First, you install the service definition into the WLM couple data set so that it is accessible to all systems in the sysplex. Second, you activate one of the service policies from the definition.

WLM Manages Work in Sysplex to Achieve Your Goals

- WLM manages competing workloads throughout the sysplex via:
  - Workload distribution
  - Workload balancing
  - Resource distribution
- In order to meet performance goals and business objectives
- As defined in a “service definition”

Installations today process different types of work with different response times. Every installation wants to make the best use of its resources and maintain the highest possible throughput and achieve the best possible system responsiveness.

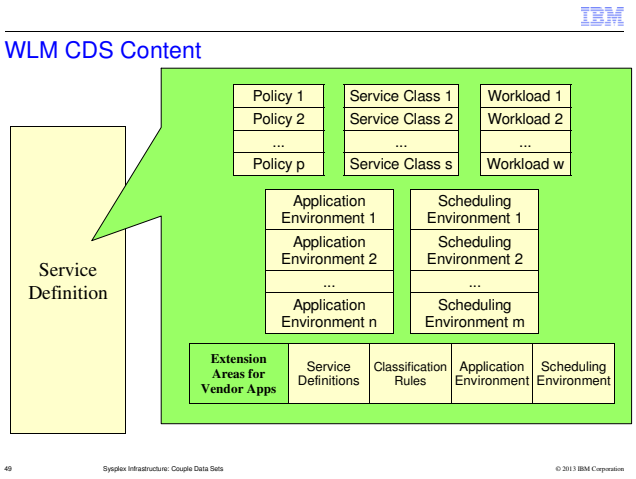
With workload management, you define performance goals and assign a business importance to each goal. You create a service definition to define the goals for work in business terms, and the system decides how much resource, such as CPU and storage, should be given to the work to meet its goal. Workload management algorithms use the service definition information and internal monitoring feedback to check how well they are doing in meeting the goals. The algorithms periodically adjust the allocation of resource as the workload level changes. Reporting reflects how well the system is doing compared to its goals.

In addition to internal feedback monitoring, workload management keeps track of what is happening in the sysplex in the form of real time performance data collection, and delay monitoring. All this information is available for performance monitors and reporters for integration into detailed reports.

For each system, workload management handles the system resources. Workload management coordinates and shares performance information across the sysplex. How well it manages one system is based on how well the other systems are also doing in meeting the goals. If there is contention for resources, workload management makes the appropriate trade-offs based on the importance of the work and how well the goals are being met.

Workload management can dynamically start and stop server address spaces to process work from application environments. Workload management starts and stops server address spaces in a single system or across the sysplex to meet the work's goals.

You can turn over management of batch initiators to workload management, allowing workload management to dynamically manage the number of batch initiators for one or more job classes to meet the performance goals of the work.



The slide depicts content of service definition installed in the WLM Couple Data Set:

One or more **service policies**, which are named sets of overrides to the goals in the service definition. When a policy is activated, the overrides are merged with the service definition. You can have different policies to specify goals intended for different times. Service policies are activated by an operator command, or through the ISPF administrative application utility function.

Workloads aggregate a set of service classes together for reporting purposes. Service classes, which are subdivided into periods, group work with similar performance goals, business importance, and resource requirements for management and reporting purposes. You assign performance goals to the periods within a service class.

Service classes, which are subdivided into periods, group work with similar performance goals, business importance, and resource requirements for management and reporting purposes. You assign performance goals to the periods within a service class.

Report classes group work for reporting purposes. They are commonly used to provide more granular reporting for subsets of work within a single service class.

Resource groups define processor capacity boundaries across the sysplex. You can assign a minimum and maximum amount of CPU service units per second to work by assigning a service class to a resource group.

Classification rules determine how to assign incoming work to a service class and report class.

Application environments are groups of application functions that execute in server address spaces and can be requested by a client. Workload management manages the work according to the defined goal, and automatically starts and stops server address spaces as needed.

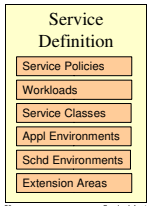
Scheduling environments are lists of resource names along with their required states. If an MVS image satisfies all of the requirements in a scheduling environment, then units of work associated with that scheduling environment can be assigned to that MVS image.

Extension Areas are for use by system management product vendors who wish to include some of their own unique information about customer workload definitions along with the WLM definitions.

Couple Data Set Format Utility … WLM CDS

```
DATA TYPE(WLM)
     ITEM NAME(POLICY)    NUMBER(#workload policies)
     ITEM NAME(WORKLOAD)  NUMBER(#workloads)
     ITEM NAME(SRVCLASS)  NUMBER(#service classes)
     ITEM NAME(APPLENV)   NUMBER(#application environments)
     ITEM NAME(SCHENV)    NUMBER(#scheduling environments)

     ITEM NAME(SVDEFEXT)  NUMBER(#KB for service definitions)
     ITEM NAME(SVDCREXT)  NUMBER(#KB for classification rules)
     ITEM NAME(SVAEAEXT)  NUMBER(#KB for application env area)
     ITEM NAME(SVSEAEXT)  NUMBER(#KB for scheduling env area)
```



- You can use IXCL1DSU , but ...
- The WLM Administrative Application provides support to create the WLM CDS:
  - Allocate couple data set
  - Allocate couple data set using CDS values

See SYS1.SAMPLIB(IWMFTCDS)

The intended users of SVDEFEXT, SVDCREXT, SVAEAEXT, and SVSEAEXT are system management product vendors who wish to include some of their own unique information about customer workload definitions along with the WLM definitions. The WLM interfaces allow these extensions to accompany the service class definitions, report class definitions, or even classification rules. The amount of extra information is specific to each product that exploits these interfaces. The product documentation should indicate how to set SVDEFEXT, SVDCREXT, SVAEAEXT, and SVSEAEXT to ensure that there is sufficient space available in the WLM couple data set to hold the extra information.

WLM Administrative Application

Allocate couple data set

Use this option to allocate both your primary and alternate WLM couple data sets. This option is for users who are allocating a WLM couple data set for the first time. This function fills in the size parameters based on the service definition you are about to install. It gets the values from the service definition data set containing the service definition.

Allocate couple data set using CDS values

Use this option to allocate both your primary and alternate WLM couple data sets based on your existing WLM couple data set size. The application displays the current size values on the panel. This function primes the size parameters with values from the current couple data set. This way, you can ensure that the new couple data set is at least as large as the current one. If the new one is smaller than the current one, you will not be able to print it into use by the sysplex. You may need to increase one or more of the primed values if you have added new objects to your service definition since the last time you installed it.

Creating WLM CDS Content

- You must
  - Format the WLM CDS
  - Use tools to create one or more service definitions
  - Use tools to install the service definition in the CDS for use by the sysplex
  - Optionally activate a service policy in the installed service definition
- The WLM CDS contains data from different sources
  - Service Definition — Static data set when "installed"
  - Active service policy — Designation set when policy "activated"
- Your influence on the service definition is limited to the exact point in time when installed in the WLM CDS (or when activating policy)
  - When you install a service definition, you copy it into the CDS from the service definition data set
  - When you update the service definition, you update the copy that resides in the service definition data set. The installed copy in the CDS is not changed.
  - Sysplex only looks at the installed copy.

You can work in the ISPF administrative application with one service definition at a time. In order to make the service definition accessible to all systems in the sysplex, you store the service definition on a WLM couple data set. Only one service definition can be installed on the WLM couple data set at a time.

If you want to work on more than one service definition at a time, you can keep each in a distinct MVS partitioned data set (PDS), or in an MVS sequential data set (PS). As an MVS service definition data set, the service definition is subject to all the same functions as an MVS data set. You can restrict access to the service definition data set, send it, and copy it, as you can any MVS data set.

The service definition must be installed on a WLM couple data set, and a service policy activated. Only service policies in the service definition installed on the WLM couple data set can be activated.

When you set up your service definition, you identify the workloads, the resource groups, the service classes, the service class periods, and goals based on your performance objectives. Then you define classification rules and one or more service policies. This information makes up the base service definition.

A two-step process is required before the sysplex starts using a new service definition. First, you install the service definition onto the WLM couple data set. Second, you activate one of the service policies from the definition.

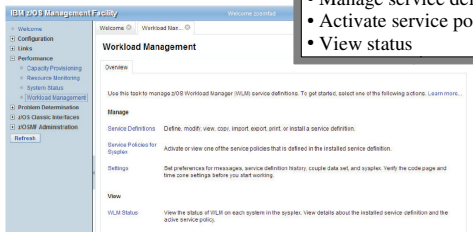
Note to WLM Users: The administrative data utility does not support WLM policy information. Those policies are stored in a formatted WLM couple data set either with an ISPF application of a z/OSMF task. See *z/OS MVS Planning: Workload Management* for a description.

Creating WLM Policies (Service Definitions)

The Data Utility IXCMIAPU is not used for WLM

- z/OS WLM Administrative Application
  - Also called the "WLM ISPF application"
  - EX 'SYS1.SBLSCLI0(IWMARIN0)'
- z/OSMF

Either one can be used to:
- Manage service definitions
- Activate service policy
- View status



WLM Administrative Application

An ISPF application that can be used to manage service definitions, create WLM couple data sets, install a service definition, and activate a service policy.

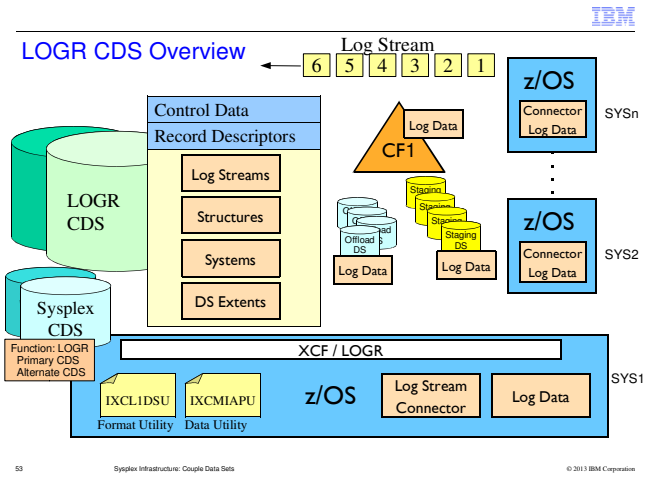
z/OSMF

The Workload Management task in z/OS Management Facility (z/OSMF) provides a browser-based user interface that you can use to manage z/OS Workload Manager (WLM) service definitions and provide guidelines for WLM to use when allocating resources. Specifically, you can define, modify, view, copy, import, export, and print WLM service definitions. You can also install a service definition into the WLM couple data set for the sysplex, activate a service policy, and view the status of WLM on each system in the sysplex.

Actions that require the Workload Management task to interact with the sysplex are limited to the sysplex in which the z/OSMF host system is a member. Such actions include installing a service definition, activating a service policy, viewing the sysplex status, and so on. If you want to interact with another sysplex, z/OSMF must be installed on a system in that sysplex and you must log into that z/OSMF instance. You can use the service definition import and export functions to copy a service definition from one z/OSMF instance to another z/OSMF instance.

Service Policies

When you create the service definition, there is an implied policy. You might define additional service policies to provide, for example, different performance goals at different times of the day. You would then activate the alternate service policy at an appropriate time. You can activate a service policy from within the application, or you can enter the command: VARY WLM,POLICY=*polname*



System Logger is an MVS component that provides a logging facility for applications. Logger saves the log data with the requested persistence, retrieves it, archives it, and deletes it when expired. Logger provides the ability to have a single merged log containing log data from multiple instances of an application within the sysplex.

When an application passes log data to System Logger, the data can initially be stored on DASD, in what is known as a DASD-only log stream, or it can be stored in a Coupling Facility (CF) in what is known as a CF-Structure log stream. In a **CF log stream**, interim storage for log data is in CF list structures. This type of log stream supports the ability for exploiters on more than one system to write log data to the same log stream concurrently. In a **DASD-only log stream**, interim storage for log data is contained in a data space in the z/OS system. The data spaces are associated with the System Logger address space, IXGLOGR. DASD-only log streams can only be used by exploiters on one system at a time. The location of the log data is transparent to the application; all log records appear as if kept in a single file of limited size. Logger manages the placement of the data to provide optimal performance while maintaining the integrity of the data.

Interim Storage is the primary storage used to hold log data that has not yet been offloaded. The interim storage medium used depends on how the log stream has been defined; it may be a Coupling Facility (CF) structure or a staging data set. Log data that is in interim storage is duplexed to prevent against data loss conditions. The data is usually duplexed to a data space, although log streams residing in a CF structure may optionally be duplexed to a staging data set.

Staging data sets are interim storage on DASD used to duplex log data that has not yet been offloaded to offload data sets. Staging data sets are required for DASD-only log streams, and are optional for CF-Structure based log streams.

Offload data sets are auxiliary storage on DASD for log streams, sometimes also referred to as log data sets or log stream data sets; they are single extent VSAM linear data sets. When the interim storage for a log stream is filled to its high offload threshold point, the System Logger begins offloading data from interim storage to the offload data sets.

Log stream definitions describe the attributes of a log stream. They are defined using IXCMIAPU and stored in the **LOGR Couple Data Set** (CDS). The CDS must be accessible to all systems in the sysplex. The CDS also contains status information about all the log streams currently in use.

System Logger

- Provides common set of logging services for applications to:
  - Save data
  - Archive data
  - Retrieve data
  - Delete expired data
- Enables multiple instances of an application scattered throughout the sysplex to have one common log

System Logger is an MVS component that provides a logging facility for applications running in a monoplex or multi-system sysplex. The advantage of using System Logger is that the responsibility for tasks such as saving the log data (with the requested persistence), retrieving the data (potentially from any system in the sysplex), archiving the data, and expiring the data is removed from the creator of the log records. In addition, Logger provides the ability to have a single, merged, log, containing log data from multiple instances of an application within the sysplex.

Log data managed by the System Logger may reside in processor storage, in a Coupling Facility structure, on DASD, or potentially on tape. However, regardless of where System Logger is currently storing a given log record, from the point of view of the exploiter, all the log records are kept in a single file that is a limited size. The location of the data, and the migration of that data from one level to another, is transparent to the application and is managed completely by System Logger, with the objective of providing optimal performance while maintaining the integrity of the data.

By providing these capabilities using a standard interface, many applications can obtain the benefits that System Logger provides without having to develop and maintain these features themselves. This results in faster development, more functionality, and better reliability. Enhancements to System Logger, such as support for System Managed CF Structure Duplexing, become available to all System Logger exploiters as soon as they are implemented in System Logger, rather than having to wait for each exploiter to design, write, and test their own support.

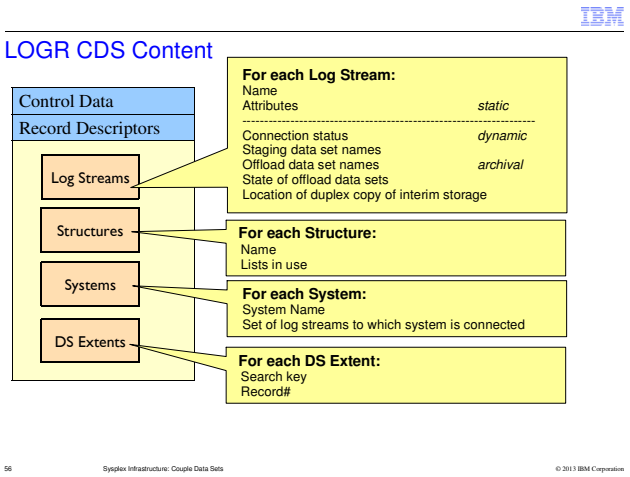
Logger Policy (or Logger Inventory)

- Defines
  - Characteristics of each log stream
  - Coupling facility structure to be used when there are active connections to the log stream
- Which in turn determines how Logger will manage the log stream
- Contains transient status information such as log stream connectors
- As well as persistent status information needed when applications need to recover from their logs
  - Data to construct the log stream from the various data sets in which the log data might reside

There are basically two types of users of System Logger. Some exploiters basically use System Logger as an archival facility for log data. These exploiters dump their log data into System Logger and rely on it to manage the archival and expiration of the data from there on. Of course, these exploiters have the ability to subsequently retrieve the data should they need to do so, but their normal mode of operation would be to just give data to System Logger and not look for it back again. And example of this type of exploiter would be the CICS Forward Recovery logs, where CICS stores data away in case a forward recovery is required some time in the future.

The other type of exploiter typically uses the data more actively, and explicitly deletes it when it is no longer required. An example of this would be the CICS DFHLOG. CICS stores information in DFHLOG about running transactions, and deletes the records as the transactions complete.

As you can imagine, the performance requirements of these exploiters will differ. The exploiters that use Logger primarily to archive data are not particularly concerned about retrieval speeds, whereas an active user of the data will ideally want all the active data to be kept in a high performance location, like a data space or a CF structure.



The System Logger policy is different than most other z/OS policy definitions in that you can only have one per sysplex. Additionally, whereas most other CDSs only contain static policy statements and some transient information, the LOGR CDS contains persistent information, such as the names of the offload data sets associated with each log stream.

Specifically, the System Logger policy contains:

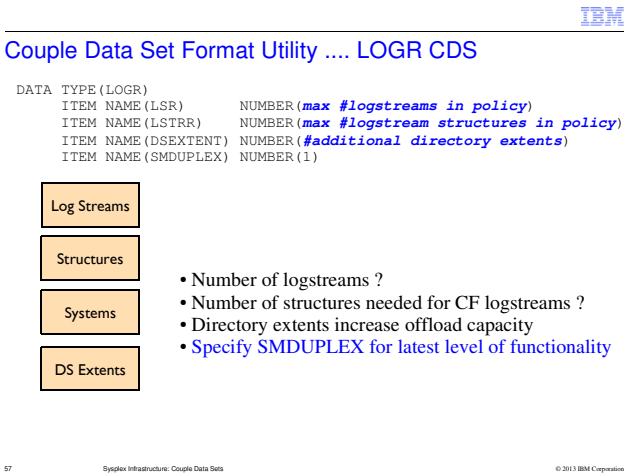
CF structure definitions

Log stream definitions

Data reflecting the current state of all log streams (connection status, names of staging and offload data sets, High RBA and log block BLOCKIDs for each offload data set, location of the duplex copy of the interim storage, and so on)

A list of the connections for every system

A count of the number of active systems in the sysplex



LSR
The maximum number of log streams that can be defined in the System Logger policy that will be stored in this CDS. The default is 1, the minimum is 1, and the maximum is 32767. Do not take the default on this parameter or you will be limiting your sysplex to one log stream. You should evaluate the System Logger applications you plan to use and determine the number of log streams needed by each (keeping in mind some applications that run on separate systems may require a single sysplex-wide log stream; others might use separate log streams for each system).

LSTRR
The maximum number of structure names that can be defined in the System Logger policy. The default is 1, the minimum is 1, and the maximum is 32767. If you plan on using only DASD-only log streams, it is not necessary to specify this parameter. If you are planning on using CF-Structure based log streams, you should determine how many structures are necessary.

DSEXTENT
The number of additional offload data set directory extents available. The default is 0, the minimum is 0, and the maximum is 99998. Each DSEXTENT (directory extent) goes into a common pool available to any log stream in the sysplex, and is allocated as needed. If you have log streams that will exceed 168 offload data sets (for example, if you retain log data for long periods of time) you should specify this parameter. Log streams start with a base of up to 168 directory entries (that is, 168 offload data sets that can be used, 1 per directory entry); each additional DSEXTENT specified will allow the log stream owning the extent to use an additional 168 directory entries.

SMDUPLEX
Ostensibly SMDUPLEX indicates whether Logger should *support* system-managed duplexing rebuild. It does not imply that Logger will *exploit* system managed rebuild. Thus there is no down side to specifying SMDUPLEX. In fact, since SMDUPLEX affects the LOGR CDS format level (version), there is a down sides to *not* specifying SMDUPLEX. If your LOGR CDS is not at the latest format level, there are several desirable Logger functions and behaviors that will not be available. Thus SMDUPLEX should be specified.

Creating LOGR CDS Content

- You must
  - Format the LOGR CDS
  - Create one or more log stream definitions in the CDS

Nothing needs to be explicitly started

- The LOGR CDS contains data from different sources
  - Log stream definitions — Static data set by Data Utility (you)
  - CF Structures — Static data set by Data Utility (you)
  - Log stream status — Dynamic data set by systems in the sysplex
- You influence the “active policy” every time the utility is run
  - You change the log stream definitions being used by the sysplex
  - Some changes to a log stream will be rejected if the log stream is in use
  - The accepted changes are immediately visible to the sysplex

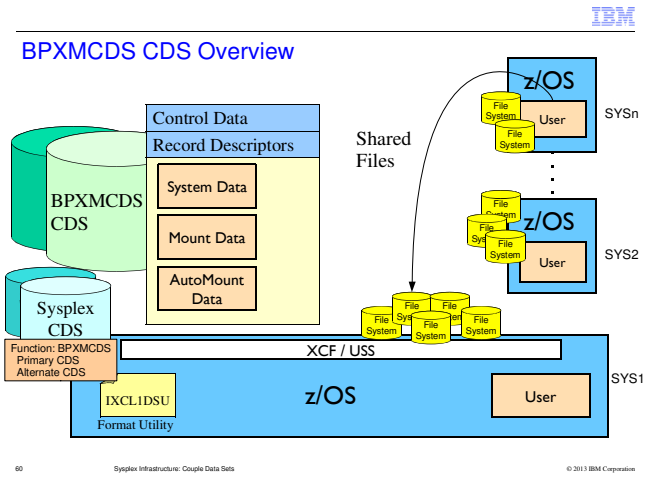
Not at all like CFRM, ARM, and SFM

Creating LOGR “Policies”

- The data utility program IXCMIAPU is used to manipulate log stream definitions in a formatted LOGR Couple Data Set
- The parameters you specify define
  - The log stream characteristics
  - The coupling facilities to be used
- Which in turn determines how Logger will manage the log stream
- The data utility can
  - Create new log stream definitions
  - Update log stream definitions
  - Delete log stream definitions
  - Report on the log streams currently defined in the CDS
- You might run the utility to manipulate an individual log stream
  - Perhaps a new application

The System Logger component manages log streams based on policy information that you place in the LOGR CDS. This can be a point of confusion, as you will often see references to the System Logger policy, the LOGR CDS (also called the System Logger CDS), or the System Logger inventory; these are not separate entities. While some other components of z/OS have multiple policies (CFRM and SFM, for example), the LOGR CDS contains the only System Logger policy. The System Logger policy contains CF structure definitions (if applicable), log stream definitions, and data describing the current status of all log streams. To understand and effectively manage System Logger, it is important to remember this difference between the LOGR CDS and the other sysplex CDSs.

In a sense, the LOGR CDS is effectively one big active policy that does not contain any administrative policies. WIth the other types of policy based function CDS, an administrative policy would be created, replace, or deleted by IXCMIAPU but the active policy would not be be touched. With a LOGR CDS there are no administrative policies and IXCMIAPU is in effect making incremental changes to the content of the active policy – the one and only policy in the CDS.



By establishing the shared file system environment, sysplex users can access data throughout the file hierarchy from any system in the sysplex. With shared file system support, all file systems mounted by a system participating in a shared file system are available to all participating systems. In other words, once a file system is mounted by a participating system, that file system is accessible by any other participating system.

The TYPE(BPXMCDS) couple data set (CDS) contains the sysplex-wide mount table and information about all participating systems, and all mounted file systems in the sysplex.

The first system that enters the sysplex with SYSPLEX(YES) initializes the CDS for z/OS UNIX System Services. The z/OS UNIX CDS controls shared file system mounts and will eventually contain information about all systems participating in the shared file system configuration.

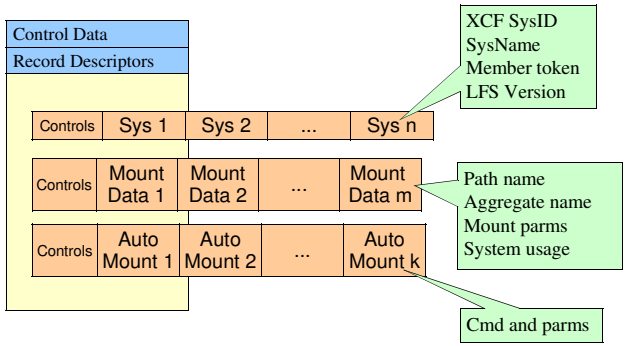
This system processes its BPXPRMxx parmlib member, including all its ROOT and MOUNT statement information. The MOUNT and ROOT information are logged in the CDS so that other systems that eventually join the participating group can read data about systems that are already using shared file system.

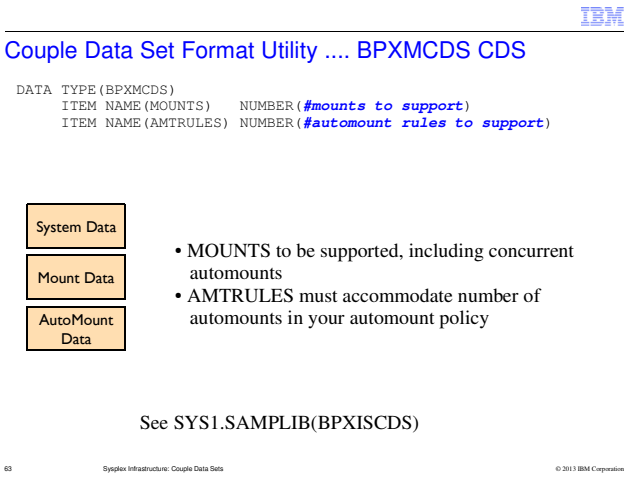
Subsequent systems joining the participating group will read what is already logged in the CDS and will perform all mounts. Any new BPXPRMxx mounts are processed and logged into the CDS. Systems already in the participating group will then process the new mounts added to the CDS.

BPXMCDS Function Supports Shared File System

- Shared File System
  - A file mounted on one system can be accessed by a different system without the user needing to log onto the other system
- The BPXMCDS Function CDS enables this support by providing a central repository of information about which files are mounted to what system

BPXMCDS CDS Content





To allocate and format a TYPE(BPXMCDS) CDS, customize and invoke the BPXISCDS sample job in SYS1.SAMPLIB. The job will create two couple data sets: one is the primary and the other is a backup that is referred to as the alternate. In BPXISCDS, you also specify the number of mount records that are supported by the CDS.

MOUNTS specifies number of mounts to be supported.

AMTRULES specifies the number of automount rules that can be supported by z/OS UNIX. It must be large enough to describe each automount-managed directory in your automount policy.

Automount mounts must be included in the MOUNTS value. The number of automount mounts is the expected number of concurrently mounted file systems using the automount facility. For example, you might have specified 1000 file systems to be automounted, but if you expect only 50 to be used concurrently, then factor these 50 into your MOUNTS value.

Creating BPXMCDS CDS Content

- Nothing for you to do beyond:
  - Formatting the CDS
  - Bringing CDS into service
  - Specifying SYSPLEX(YES) in BPXPRMxx parmlib member
    - System will be part of file sharing group
    - Other customization likely desirable as well
- All data is updated dynamically by those systems in the sysplex that participate in the file sharing group

Checkpoint

- We have looked at
  - Use and content of each type of CDS
  - Format utility parameters to create each kind of CDS
  - How data gets into the CDS
- We return to the format utility to consider
  - Other parameters and options
  - Which leads to questions of CDS placement ...

Couple Data Set Format Utility … Reprise

- The format utility program IXCL1DSU creates and formats a Couple Data Set (CDS) using parameters that you specify
- These parameters determine name and placement of the CDS

```
//FMTCDS   JOB  ....
//STEP1    EXEC PGM=IXCL1DSU
//STEPLIB  DD   DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
   DEFINEDS SYSPLEX(name of sysplex)
            DSN(cds data set name) VOLSER(volume)
            MAXSYSTEM(number of systems)
         DATA TYPE(type name)
            ITEM NAME(record name) NUMBER(count)
            ITEM NAME(record name) NUMBER(count)
              ....
```

One DEFINEDS for each CDS. How many ?

DEFINEDS keywords ?

One or many DATA TYPE per CDS ?

The name of the XCF couple data set format utility is IXCL1DSU. This program resides in SYS1.MIGLIB (which is logically appended to the LINKLIST). The utility is available through STEPLIB, which allows you to run older versions of the utility if you want (for example, from previous z/OS releases).

IXCL1DSU allows you to format all types of couple data sets for your sysplex. The utility contains two levels of format control statements. The primary format control statement, DEFINEDS, identifies the couple data set being formatted. The secondary format control statement, DATA TYPE, identifies the type of data to be supported in the couple data set — Sysplex Couple Data Set (SYSPLEX), Automatic Restart Management (ARM) data, Coupling Facility Resource Management (CFRM) data, Sysplex Failure Management (SFM) data, Workload Manager (WLM) data, z/OS System Logger (LOGR) data, or z/OS UNIX System Services (BPXMCDS) data.

For each DATA TYPE, you specify ITEM statements to identify the type and quantity of data to be supported by the relevant function. The particular ITEM statements that apply to any given function are documented in *z/OS MVS Settting Up a Sysplex (SA22-7625).* You will need to determine whether a given DATA TYPE is needed for a given sysplex, and if so, what type of data is needed (ITEM NAME) and how much (NUMBER).

CDS Format Utility … One or Many Functions per CDS?

- A function CDS can contain multiple functions (types)
  - For example, could put ARM and SFM in the same CDS
  - Implies fewer CDS to manage
  - XCF still manages the types independently
- No compelling technical argument for either case with respect to
  - Performance (though could depend on workload)
  - Failure processing
- So do what you find easiest to manage with regard to:
  - Using format utility to create CDS
  - Using data utility to populate function CDS with data
  - Naming conventions
  - Operational procedures
  - Disaster Recovery
- Most installations seem to prefer one function per CDS

Input to IXCL1DSU:

```
DEFINEDS SYSPLEX(name of sysplex)
    DSN(cds data set name) VOLSER(volume)
    MAXSYSTEM(number of systems)
    DATA TYPE(type name1)
         ITEM NAME(record name) NUMBER(count)
         ITEM NAME(record name) NUMBER(count)
    DATA TYPE(type name2)
         ITEM NAME(record name) NUMBER(count)
         ITEM NAME(record name) NUMBER(count)
    <repeat DATA statements for as many functions to be in this CDS>
```

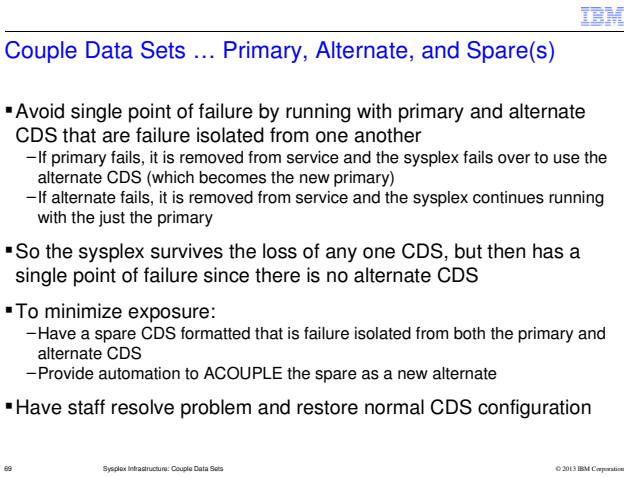
You can do some rather creative (confusing?) function CDS configurations if you have multiple types in a given function CDS. For example, you could define CDS1 as the primary CDS for both ARM and SFM but have CDS2 serve as the alternate for ARM and CDS3 serve as the alternate for SFM. Or you could have CDS1 be the primary for ARM and CDS2 be the primary for SFM, but have CDS3 serve as the alternate for them both. I'm inclined to think doing so would be terribly confusing. So if you do put multiple types in one CDS, I suspect you ought to have one CDS serve as the alternate for those types.

When XCF removes a CDS from service, it does so one type at a time. So if a primary CDS containing multiple types fails, the sysplex wide “remove CDS” protocol will be driven once for each type – even though it is the same one CDS that failed. Furthermore, initiating removal of a CDS for one type does not imply that it will be removed from service for a different type in that same CDS. A transient I/O error might induce PSWITCH for one particular type. If the errors were never observed for another type, it would happily continue using the CDS.

Couple Data Set Format Utility … DEFINEDS Parameters

- SYSPLEX
  - Name of the sysplex that is to use the CDS
  - Given sysplex will not use the CDS if the formatted name is not the same as the sysplex name
- MAXSYSTEM ← Use the same value for all your CDS
  - Number of systems in the sysplex
  - Likely same value for all CDS
- DSN
  - Traditional MVS data set name
  - The CDS must not already exist, it is to be newly created
  - Establish naming conventions for your installation
- CDS Placement
  - VOLSER, optionally with UNIT
  - STORCLAS
  - MGMTCLAS

  CDS can reside on SMS managed volumes, but need appropriate options
- CATALOG | NOCATALOG



XCF permits use of a primary and alternate for every type of couple data set (CDS) that is supported. If the primary CDS becoming unavailable for any reason, XCF automatically switches to the corresponding alternate CDS. The alternate is in effect promoted to be the new current primary CDS. If the alternate CDS becomes unavailable, it is removed from service. XCF issues a message to so indicate:

```
IXC263I REMOVAL OF THE <PRIMARY|ALTERNATE> COUPLE DATA SET dsname
IS COMPLETE.
```

However, if an existing CDS is removed from service, the sysplex is now operating with only a primary CDS, and therefore has a single point of failure. XCF issues a message to so indicate:

```
IXC267E PROCESSING WITHOUT AN ALTERNATE COUPLING DATA SET. ISSUE
SETXCF COMMAND TO ACTIVATE A NEW ALTERNATE.
```

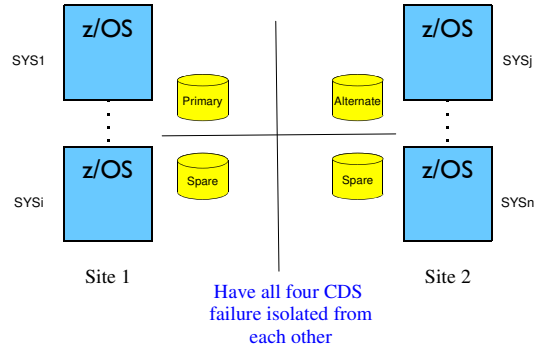
Ideally you will have automation to resolve this situation as quickly as possible. The automation should issue an appropriate command to bring a new CDS into service as the alternate CDS, thereby eliminating the single point of failure:

```
SETXCF COUPLE,TYPE=typename,ACOUPLE=(cdsname,volser)
```

Format a spare (third) CDS for the automation to ACOUPLE. I suppose you could have the automation submit an appropriate format job as needed, but that would take time and elongate the exposure to the single point of failure. The challenge is to discover the appropriate *typename* since it does not appear in the XCF messages. You would need to establish a means by which to map a given CDS name to the appropriate *typename. A* convention that includes the type as part of the CDS name is one possible way.

Someone should be charged with the task of investigating the failure that precipitated the removal of the CDS, resolving the problem, and restoring the normal CDS configuration.

CDS Configuration for Multi-Site Sysplex



In a multi-site sysplex you will likely have the primary CDS in one site and the alternate CDS in the other. Each site will also have its own spare CDS. All of these various CDS should be failure isolated from one another.

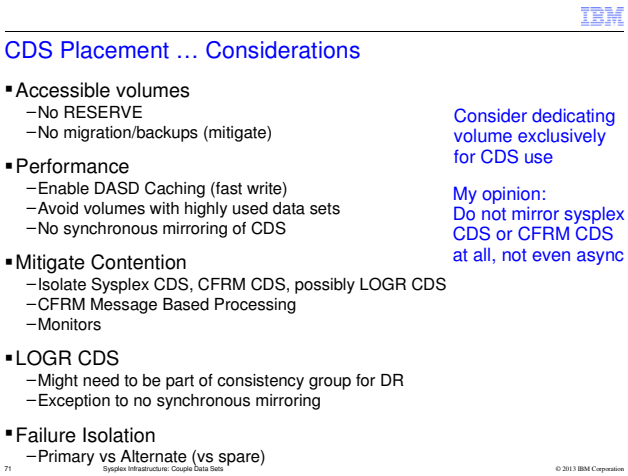
Suppose the primary CDS fails. XCF will automatically fail-over to using the alternate CDS as the new primary. The spare CDS at Site 1 should then be brought into service as the new alternate CDS.

Supposed the alternate CDS fails. The spare CDS at Site 2 should then be brought into service as the new alternate CDS.

Suppose Site 2 fails. The spare CDS at Site 1 should then be brought into service as the new alternate CDS.

Suppose Site 1 fails. XCF will automatically fail-over to using the alternate CDS at Site 2 as the new primary. The spare CDS at site 2 should then be brought into service as the new alternate CDS.

In all cases, we are able to restore redundancy and maintain failure isolation.



Separate busy CDSs from each other
While the CFRM and Sysplex CDSs are typically not very busy during normal operation, they can become very busy during recovery from a failure. As a result, placing the primary Sysplex and CFRM CDSs on the same volume can cause performance problems and increase recovery times. To ensure optimal performance, allocate these CDSs on separate volumes.

Keep an eye on the LOGR CDS. Usage will likely vary according to workload. It may be necessary to separate it from other CDS if it is heavily used at your installation.

Implement CFRM MSGBASED processing
z/OS 1.8 introduced a new CFRM protocol called MSGBASED processing. When MSGBASED processing is enabled, the communication associated with recovery or rebuild processing for most CF structures is routed via XCF signalling rather than via the CFRM CDS. This results in much better scalability for the recovery of structures by reducing sysplex wide I/O contention on the CFRM CDS.

CDS Considerations ... Restrictions

- Multiple extents not supported
- Cannot span multiple volumes
- Can only be used by one sysplex

Format Utility and Data Utility Jobs

- IXCL1DSU lets you format multiple CDS in one job
  - Many of one type
  - Even multiple types
- My suggestion:
  - One type of CDS per job
  - Always format primary, alternate, and spares together
- For both utilities:
  - It may be months or years between uses
  - You need to know where these jobs reside
  - What the parameters should be if (when?) you can't find the jobs
- Be prepared: there will come a time when you need to recreate the CDS from scratch

Security

- Protect the CDS
  - Only XCF should be accessing these data sets
- Format Utility
  - Always creates new data sets
  - So nothing beyond normal data set naming profiles
  - Still, think about who can issue SETXCF or change COUPLExx
- Administrative Data Utility
  - Need to control use since the utility can change content of CDS in a way that could be detrimental
  - Programs using it must reside in APF Authorized library
- Some functions may have additional security profiles that need to be set up

It is the responsibility of the installation to provide the security environment for the couple data sets. Consider protecting the couple data sets with with the same level of security as the XCF address space (XCFAS).

You must control use of the IXCMIAPU utility through the z/OS Security Server, which includes RACF, or your installation's security package. The security administrator should define a resource profile equivalent to UACC NONE in the FACILITY class as indicated for each of the following functions:

•For ARM, the resource name is ‘MVSADMIN.XCF.ARM’.

•For CFRM, the resource name is ‘MVSADMIN.XCF.CFRM’.

•For LOGR, the resource name is ‘MVSADMIN.LOGR’.

•For SFM, the resource name is ‘MVSADMIN.XCF.SFM’.

Assign UPDATE access authority to users who must alter or maintain the policy; assign READ access authority to users who require reports on the policy, but who will not change the policy. These FACILITY class profiles will be used for authority checking on an ACTIVE couple data set. When a couple data set is INACTIVE, the rules for normal data set protection apply.

If using LOGR, profiles will also be needed for logstream and CF structure resources well (such as logstreams and CF structures). These profiles determine whether the user of IXCMIAPU is permitted to

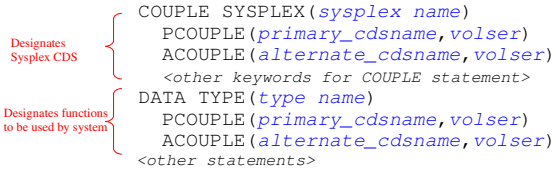
If applicable, define a resource profile for the resource name MVSADMIN.XCF.CFRM to the FACILITY class. Assign UPDATE access to users who must define coupling facility structures in the LOGR policy.

Checkpoint

- At this point, we have discussed:
  - Purpose and content of each type of CDS
  - How to create/format a CDS
  - How to work with each type of CDS
  - CDS placement
- Next, we discuss:
  - How to make CDS available to sysplex
  - How to make changes to CDS configuration

Specifying Couple Data Sets at IPL Time

- COUPLExx parmlib member



- Every system in sysplex needs to be using the same Sysplex CDS
- All systems using a given function must use the same Function CDS
  - For most practical cases, all systems will need access to the function CDS
  - Though this is not necessarily required (it depends)

Unlike the sysplex CDS, XCF does not require every system in the sysplex to have access to a given function CDS. That is, a function could be used be a subset of systems in the sysplex. However, XCF does require that all systems in the subset to use the same function CDS. Furthermore, if a system not in the subset wants to start using the function, it must use the function CDS that are already in use by the subset.

As an example, we note that the GDPS k-System will typically be configured without access to the LOGR CDS.

However, from a practical standpoint, it should be noted that many functions will not be fully functional if their function CDS is not everywhere in use by the sysplex. ARM and SFM are cases in point.

CFRM is its own special case. It is possible for CFRM to be used by a proper subset of systems in the sysplex. However, if a system connects to a CFRM CDS (PCOUPLE) but then later loses access to the CFRM CDS (both primary and alternate), the system will wait-state. Furthermore, for customers running GRS-Star, GRS requires every system in the sysplex have access to the ISGLOCK structure (out in some coupling facility). If a system IPLs into the sysplex but does not have access to the CFRM CDS, GRS will not be able to connect to the ISGLOCK structure and wait-states the system. So in practice, the CFRM function is used by ever system in the sysplex.

Dynamically Defining a Primary Function CDS

- In general, you will almost always use COUPLExx to indicate which couple data sets are to be used by a system
  - COUPLExx is the only way to get a system to use a Sysplex CDS
- You can use the SETXCF COUPLE command to dynamically define a primary function CDS for a given service in these particular two cases:
  - The function is not in use by any system in the sysplex
  - The function is in use somewhere in the sysplex, but not by this system
- SETXCF COUPLE,TYPE=*typename*,PCOUPLE=(*cdsname*,*volser*)
  - *Typename* is one of CFRM, ARM, SFM, WLM, LOGR, BPXMCDS
  - If the function CDS has multiple types, you can code TYPE=(type1,type2,...)
  - *cdsname* is name of data set formatted with IXCL1DSU
  - *volser* is optional if CDS is cataloged

The SETXCF COUPLE,PCOUPLE command specifies the data set to use as the primary couple data set for the type of service specified by TYPE. You can specify CFRM, ARM, SFM, WLM, LOGR, or BPXMCDS for TYPE. Note that you cannot specify TYPE=SYSPLEX to identify the sysplex couple data set. In order to IPL into an existing sysplex, the system must have access to the sysplex CDS being used by the rest of the systems in the sysplex. Thus the sysplex CDS can only be defined in the COUPLExx parmlib member. If a system IPLs in XCF-Local mode (that is, without a sysplex CDS), it must be re-IPLed to become a monoplex, or to start a multisystem capable sysplex, or to join an existing sysplex (all of which use a sysplex CDS). A system cannot be converted dynamically from XCF-Local mode to any other type of sysplex.

The designated data set must exist. It must have been formatted with the XCF format utility (IXCL1DSU).

If the service indicated by TYPE is already operational in the sysplex, the system ignores the data set specified by PCOUPLE. Instead, the system attempts to make the service available to the system by using the couple data set that is currently supporting the service on other systems in the sysplex. If the service is not already operational in the sysplex, the system attempts to use the specified data set as the primary couple data set for the service specified.

Note: When TYPE=CFRM is specified and the CFRM couple data set is added to the sysplex, it MUST NOT BE REMOVED OR DELETED. If the CFRM couple data set is removed from the sysplex, every system that had access to the CDS will enter a non-restartable WAIT STATE. This is true even if no CFRM policies were activated.

Beware !

- Once you PCOUPLE the CFRM CDS to a system, the system will wait-state if it loses access (to both primary and alternate)
  - Even if you never start a CFRM policy
- Loss of access to any other function CDS generally implies loss of function

Why Might CDS Configuration Need to Change?

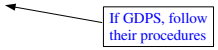
- Growth
  - CDS needs more capacity
  - Support new function
  - Support new features of an existing function
- Failure
  - Loss of either primary or alternate CDS
- Placement
  - Reconfiguring DASD
  - Reconfiguring sysplex

I did not include any information about how to determine if you are running out of room in your CDS. See Redbook "System z Parallel Sysplex Best Practices"

Splitting sysplex into two sites

Changing the CDS Configuration

- Your toolkit
  - Format utility IXCL1DSU
  - SETXCF COUPLE,TYPE=*typename*,ACOUPLE=(*cdsname,volser*)
  - SETXCF COUPLE,TYPE=*typename*,PSWITCH



- Do one or more of the following as needed in an appropriate order:
  - Format new CDS (one or more)
  - ACOUPLE to bring new CDS into service as alternate CDS
  - PSWITCH to make current alternate CDS become primary CDS
  - ACOUPLE to bring another alternate CDS into service to eliminate SPOF
  - Update COUPLExx parmlib member(s) to reflect new CDS configuration being used by sysplex
  - Delete old CDS if no longer needed

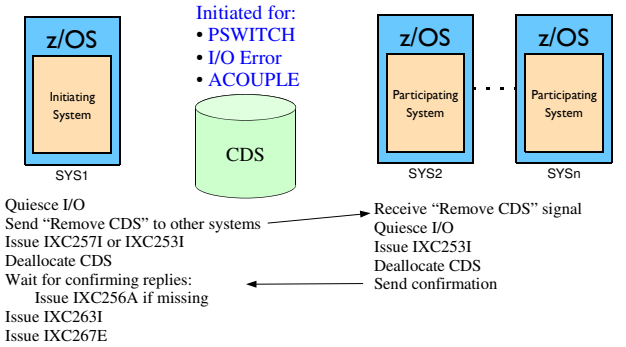
XCF Handling of Reconfiguration Processes

XCF tries to prevent local problems from cascading to the entire sysplex

APAR OA38311

- Failure of CDS
  - Probe “other” CDS to confirm viability
  - If no viable “other”, stop use of CDS locally
  - Else initiate sysplex-wide removal of failed CDS
- PSWITCH
  - Probe alternate CDS to confirm viability
  - If no viable alternate, reject command
  - Else initiate sysplex-wide removal of primary CDS
- ACOUPLE
  - If alternate CDS currently in use:
    - Probe primary CDS to confirm viability
    - If primary not viable, reject command
    - Else initiate sysplex-wide removal of alternate CDS
  - Bring new alternate CDS into sysplex-wide service

Sysplex Wide Removal of CDS From Service



IXC257I PRIMARY COUPLE DATA SET dsname1 FOR type IS BEING REPLACED BY dsname2 DUE TO OPERATOR REQUEST

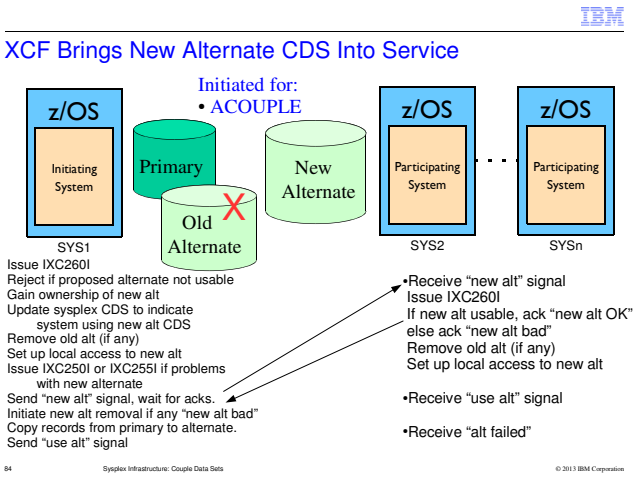
IXC253I <PRIMARY|ALTERNATE> COUPLE DATA SET dsname FOR type IS BEING REMOVED BECAUSE OF reason DETECTED BY SYSTEM sysname

IXC256A REMOVAL OF <PRIMARY|ALTERNATE> COUPLE DATA SET dsname CANNOT COMPLETE UNTIL THE FOLLOWING SYSTEM(S) ACKNOWLEDGE THE REMOVAL: sysname1 ... sysnameN

IXC263I REMOVAL OF THE <PRIMARY|ALTERNATE> COUPLE DATA SET dsname IS COMPLETE.

Sysplex Wide Removal of CDS From Service ...

- During this process, there will be no writes to either CDS
  - Neither primary nor alternate
- If a Sysplex CDS and Function CDS are being removed at the same time, removal of the Function CDS cannot complete until after removal of the Sysplex CDS completes
  - Sysplex CDS contains Function CDS configuration
  - New configuration cannot be written while sysplex CDS being removed
- Can deadlock during removal of Sysplex CDS if system fails:
  - Removal of CDS requires confirmation from each system in sysplex
  - A failed system will not respond
  - Confirmation can proceed if system removed from sysplex
  - Removal of system from sysplex requires write to Sysplex CDS
- VARY XCF,sysname,FORCE may be needed to make progress
  - Pay attention to IXC256A



IXC260I ALTERNATE COUPLE DATA SET REQUEST FROM SYSTEM sysname IS NOW BEING PROCESSED. DATA SET: dsname

New alternate is viable if it is both "suitable" and "consistent" with current primary CDS.

Suitability checks include:

- New alternate must not be same as primary
- ACOUPLE must not already be in progress
- New alternate is not same as existing alternate

Consistency checks. Relative to the current primary CDS, the proposed alternate must:

- Support all of the record types supported by the primary CDS (it can support more)
- Have space for at least as many records as the primary CDS
- Have records whose size is at least as long as the records in the primary CDS

"Extra" records are any that are present in the alternate but not in the primary. They have to be cleaned up separately because they won't be initialized by synchronization with the primary.

IXC250I ALTERNATE COUPLE DATA SET REQUEST FAILED FOR DATA SET dsname FOR type: reason

IXC255I UNABLE TO USE DATA SET dsname AS THE PRIMARY | ALTERNATE FOR type: reason

Reconfiguring Due To Growth

- Assume sysplex using OLD1, OLD2, and old spares
- Format new primary, alternate and spares with larger size
  - NEW1, NEW2, and new spares
- SETXCF COUPLE,TYPE=typename,ACOUPLE=(NEW1,volser)
- SETXCF COUPLE,TYPE=typename,PSWITCH
  - At this point, can no longer ACOUPLE any of the old CDS
- SETXCF COUPLE,TYPE=typename,ACOUPLE=(NEW2,volser)
- Update COUPLExx to reflect new CDS configuration
- Delete OLD1, OLD2, and old spares

Automation ?
GDPS ?

Automation Concerns

If you have automation dynamically ACOUPLE a spare CDS when the sysplex is running without an alternate CDS, there are some concerns. The PSWITCH will create a single point of failure that the automation might try to resolve. However, the PSWITCH brings the new larger CDS into service. At that point, any new alternate CDS must be at least as big as the newly promoted primary. If your automation is still pointing at the old smaller spares, the ACOUPLE of a smaller CDS will fail. So you will need to update the automation to ACOUPLE the new larger spare CDS.

CDS Coexistence and Versions

- When formatting a new CDS, you might make changes that:
  - Increase the number or size of existing records beyond some supported max
  - Define a new type of record
  - Require use of a new sysplex-wide protocol
- Such a CDS might not be compatible for use with down-level software
  - Intermixing software levels could lead to hangs, data corruption, etc
- Data is inserted into the CDS to identify its version or level
  - By XCF and/or the exploiting function at format time
  - By the exploiting function at run time
- Down-level software will not use an up-level CDS
  - XCF and/or the exploiting function checks the CDS version or level, and
  - Rejects use of a CDS that is not supported

XCF does not allow you to install an uplevel couple data set on a system in a sysplex in which one or more systems do not include the code for a newer version of the format utility (and for the higher maximum values specified in the utility).

When defining your primary and alternate couple data sets, the system does not allow you to mix couple data sets formatted at different versions when running in a sysplex in which one or more systems do not include the code for a newer version of the format utility (and for the higher maximum values specified in the utility).

CDS Coexistence and Versions … Implications

- Once an up-level CDS becomes the primary CDS, you can't easily fall back to a down-level CDS
  - Requires re-IPL of sysplex or function outage
  - Re-IPL of sysplex will likely need newly formatted sysplex CDS to ensure that there are no references to the new no longer desired function CDS
- If the up-level CDS is already in use by the sysplex, a down-level system might not be able to IPL into the sysplex
  - For Sysplex CDS and CFRM CDS
  - If otherwise makes it in, will not be able to use the function
- If a down-level system exists in the sysplex, an up-level CDS cannot be brought into service
  - PCOUPLE on down-level system will fail
  - ACOUPLE certainly fails if down-level system is using function
  - ACOUPLE might fail if down-level system is in the sysplex, whether using the function or not

CDS Coexistence and Versions … Possible Upgrade Path

- Roll compatibility support for the up-level CDS around the sysplex
  - Generally tolerates new level but does not exploit functionality
  - Prevents damage to data later used by up-level support
- Format and ACOUPLE up-level CDS
  - Compatibility support allows up-level CDS to be brought into service
  - At this point you can still ACOUPLE a down-level CDS
- PSWITCH to make up-level CDS be primary CDS
  - You can no longer fall back to a down-level CDS
  - Any new ACOUPLE must be done with up-level CDS
- Roll exploitation support for the up-level CDS around the sysplex, enabling exploitation as applicable
  - Perhaps as each system becomes up-level
  - Perhaps after up-level support rolled around the sysplex
  - Perhaps application does so automatically when capable

CDS Format Time and CDS Versions

- CDS versions and levels are not changed all that often
  - Usually on a release boundary
  - In support of some new feature
- DISPLAY XCF,COUPLE,TYPE={typename|ALL}
  - TOD when CDS formatted
  - Usually “Additional Information” to indicate what CDS supports
    - Not WLM. Use DISPLAY WLM to see “WLM CDS Format Level”
    - Info often correlates to format utility input, but not necessarily
    - Refer to function documentation for interpretation
    - Probably ought to use latest CDS version after necessary support is rolled around the sysplex
- The important aspect for the sysplex is whether the desired attributes, features, or functions are supported by the primary CDS
  - An old format TOD is OK
  - So long as the CDS has the desired support

Sometimes ACOUPLE and IXCMIAPU Confusion

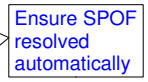
- For example:
  - Format new CDS
  - Run data utility to create new policies in the new CDS
  - ACOUPLE the new CDS
    - Which copies content of current primary CDS to the new alternate
  - Oops, lost the new policies that were put in the new CDS
- When using the data utility, you either:
  - Manipulate administrative policies in primary CDS currently used by sysplex
  - Manipulate administrative policies in an off-line CDS that is to become the primary CDS for:
    - A new sysplex, or
    - An existing sysplex that has never before used the function
    - An existing sysplex for which you somehow made it through the difficult task of eliminating use of the function CDS

IXCMIAPU and Target Function CDS

- DSN and VOLSER keywords determine what CDS will be processed by the Administrative Data Utility
- Omit these keywords to update the current primary function CDS that is in use by the sysplex
  - As would be the case if you want to create or update an administrative policy in the CFRM, ARM, or SFM function CDS being used by the sysplex
  - In order to activate that new policy via a SETXCF START,POLICY command
- Specify these keywords to update a function CDS that is not currently in use by any sysplex
  - As would be the case if you want to:
    - Create policies for the initial IPL of a sysplex, or
    - Have a CDS containing your administrative policies for use at a DR site
  - This CDS would be used when:
    - IPLing the sysplex with freshly formatted Sysplex CDS, or
    - Doing PCOUPLE the first time a function is brought into service

Reconfigure Due to Failure

- Assume sysplex using CDS1, CDS2, and spare CDS3
- If primary CDS1 fails:
  - System automatically makes CDS2 become new primary
  - SETXCF COUPLE,TYPE=typename,ACOUPLE=CDS3
- If alternate CDS2 fails:
  - System automatically removes CDS2 from service
  - SETXCF COUPLE,TYPE=typename,ACOUPLE=CDS3
- At this point you should either:
  - Update COUPLExx to reflect new CDS configuration, or
  - Continue repair procedure to restore “normal” CDS configuration in COUPLExx



Reconfigure Due to Placement

- Assume sysplex is using CDS1, CDS2 and you want the CDS to be on different volumes
- Format CDS3, CDS4 on desired new volumes
- SETXCF COUPLE,TYPE=typename,ACOUPLE=CDS3
- SETXCF COUPLE,TYPE=typename,PSWITCH
- SETXCF COUPLE,TYPE=typename,ACOUPLE=CDS4
- Update COUPLExx to reflect new CDS configuration
- Delete CDS1, CDS2

If GDPS, use their process

Reconfigure Due to Placement … Data Migration ?

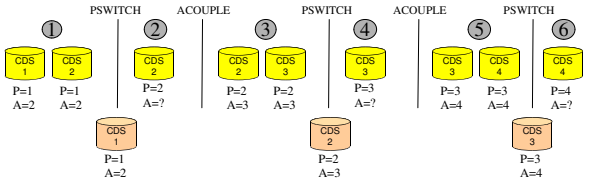
- Data migration products claim to move CDS from one volume to another transparently and non-disruptively to live sysplex
- And they almost always do so successfully
- But when they don't … disaster
  - Data in CDS is corrupted
  - Sysplex IPL with newly formatted CDS's is the only way to ensure restoration of reliable operation
- My opinion:
  - Not worth the risk
  - Always use ACOUPLE and PSWITCH to migrate CDS to new volumes
  - Never run migration against a volume with a CDS currently in use by sysplex

Checkpoint

- We can
  - Create CDS
  - Make them available for use by the sysplex
  - Change the ones being used by the sysplex
- I have repeatedly said: “Update COUPLExx to reflect the current CDS configuration being used by the sysplex”
  - Regardless of whether sysplex CDS or function CDS configuration changes
- So what happens if your CDS configuration changes and you IPL a system with a COUPLExx parmlib member that does not reflect the current CDS configuration being used by the sysplex?

Let's consider the sysplex CDS
followed by the function CDS ...

IPLing After Sysplex CDS Configuration Changes



For each case 1 to 6, IPL system with COUPLExx indicating:
PCOUPLE(CDS1)
ACOUPLE(CDS2)

```
IXC2681 "CDS in COUPLExx not consistent"
IXC275I "Here are CDS defined in COUPLExx"
IXC273I "Looking for CDS last used by sysplex"
IXC275I "Here are CDS last used by sysplex"
```

If find an active sysplex at the end of the chase, system joins sysplex.
If doubts about the state of the sysplex, the operator is prompted:

```
IXC269D REPLY U TO USE RESOLVED DATA SETS, C TO USE COUPLE DATA SETS
SPECIFIED IN COUPLExx, OR R TO RESPECIFY COUPLEXX
```


In each case, the IPLing system will try to open the sysplex couple data sets indicated in its COUPLExx parmlib member. If a couple data set (CDS) cannot be opened, it will be dropped from consideration. For each CDS that can be opened, the IPLing system will look inside the CDS and determine what CDS configuration is being used by the systems that appear to be active in the sysplex as recorded in that CDS. Note that this information could be residual data that no longer accurately reflects the state of the sysplex.

We use the notation P=CDSi to indicate that CDSi is the primary sysplex CDS and A=CDSj to indicate that CDSj is the alternate sysplex CDS.

Case 1: Sysplex using P=CDS1, A=CDS2
Both CDS1 and CDS2 show the same sysplex CDS configuration as being in use, and this configuration matches the COUPLExx specification. The system joins the existing sysplex identified by its COUPLExx. This is the best case and you should strive to maintain a COUPLExx member that allows this to happen.

The operator issues a SETXCF COUPLE,PSWITCH command or the primary CDS fails. The sysplex is now running with CDS2 as the primary but there is no alternate.

Case 2: Sysplex using P=CDS2, no alternate.
The IPLing system opens CDS1 and CDS2 indicated by COUPLExx. The IPLing system has two sysplex CDS configurations to consider. Configuration 1 seen in CDS1: (P=CDS1,A=CDS2) and configuration 2 seen in CDS2: (P=CDS2,A=none). If a sysplex had been using configuration 1, the primary and alternate CDS would both report the same CDS configuration of (P=CDS1,A=CDS2). Since CDS2 does not show that configuration, configuration 1 could not be the sysplex CDS configuration that was last used by a sysplex. On the other hand, configuration 2 seen in CDS2 is self consistent and could be a configuration that was last used by the sysplex. The COUPLExx definitions do not match those that are in effect for the sysplex. The IPLing system issues messages to document the mismatch and the CDS it is using, namely (P=CDS2,A=none).

The operator issues SETXCF COUPLE,ACOUPLE command to bring in CDS3 as an alternate CDS.

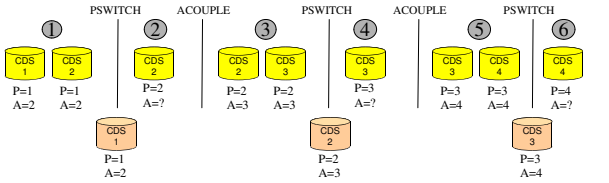
Case 3: Sysplex using P=CDS2, A=CDS3
The IPLing system opens CDS1 and CDS2. What are the possible CDS configurations? CDS1 shows (P=CDS1,A=CDS2) which is not self-consistent. CDS2 shows (P=CDS2,A=CDS3). So the IPLing system also opens CDS3 which also has a CDS configuration of (P=CDS2,A=CDS3). Thus CDS2 and CDS3 are self-consistent. The system joins the sysplex using CDS configuration (P=CDS2,A=CDS3).

The operator issues a SETXCF COUPLE,PSWITCH command or the primary CDS fails. The sysplex is now running with CDS3 as the primary but there is no alternate.

Case 4, 5, and 6: See next slide for explanatory notes

Duplicate of previous slide to continue speaker notes

IPLing After Sysplex CDS Configuration Changes



For each case 1 to 6, IPL system with COUPLExx indicating:
PCOUPLE(CDS1)
ACOUPLE(CDS2)

```
IXC2681 "CDS in COUPLExx not consistent"
IXC275I "Here are CDS defined in COUPLExx"
IXC273I "Looking for CDS last used by sysplex"
IXC275I "Here are CDS last used by sysplex"
```

If find an active sysplex at the end of the chase, system joins sysplex.
If doubts about the state of the sysplex, the operator is prompted:

```
IXC269D REPLY U TO USE RESOLVED DATA SETS, C TO USE COUPLE DATA SETS
SPECIFIED IN COUPLExx, OR R TO RESPECIFY COUPLEXX
```


Cases 1, 2, and 3 are explained in the notes for the previous slide.

Case 4: Sysplex using P=CDS3, no alternate
The IPLing system opens CDS1 and CDS2 indicated by COUPLExx. Note that neither CDS is currently in use by the sysplex. Each CDS will have residual data reflecting the state of the sysplex at the moment when the CDS was last used by the sysplex. CDS1 shows CDS configuration of (P=CDS1,A=CDS2). CDS2 shows CDS configuration of (P=CDS2,A=CDS3). Since the configuration reported by CDS1 is not self-consistent, the IPLing system considers the configuration seen in CDS2. To do that, the IPLing system must open CDS3 and examine its CDS configuration. CDS3 shows CDS configuration of (P=CDS3,A=none). Thus the configuration seen in CDS2 is not self-consistent and cannot be a configuration in use by a sysplex. However, the configuration (P=CDS3,A=none) seen in CDS3 is self-consistent. The system joins the sysplex using CDS configuration (P=CDS3,A=none).

The operator issues a SETXCF COUPLE,ACOUPLE command to bring in CDS4 as the alternate sysplex couple data set. The sysplex now has both a primary and an alternate.

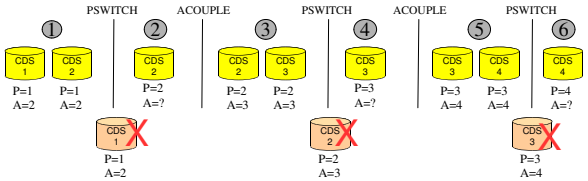
Case 5: Sysplex using P=CDS3, A=CDS4
Very similar to Case 4. The IPLing system opens CDS1 and CDS2 indicated by COUPLExx. The CDS configuration (P=CDS1,A=CDS2) seen in CDS1 is not self-consistent because CDS2 has a different CDS configuration (P=CDS2,A=CDS3). The IPLing system opens CDS3 which shows a CDS configuration of (P=CDS3,A=CDS4), which is not consistent with the configuration shown in CDS2. The IPLing system opens CDS4. The CDS configuration (P=CDS3,A=CDS4) matches the configuration in CDS3. The system joins the sysplex using CDS configuration (P=CDS3,A=CDS4).

The operator issues SETXCF COUPLE,PSWITCH or CDS3 fails so CDS4 becomes the primary sysplex CDS. The sysplex is running without an alternate.

Case 6: Sysplex using P=CDS4, no alternate
We now have a familiar pattern. The IPLing system opens CDS1 and sees CDS configuration of (P=CDS1,A=CDS2), opens CDS2 and sees (P=CDS2,A=CDS3), opens CDS3 and sees (P=CDS3,A=CDS4). Finally it opens CDS4 and sees (P=CDS4,A=none). Only the configuration seen in CDS4 could be in use by an actual sysplex. The system joins the sysplex using CDS configuration (P=CDS4,A=none).

Thus we see how an IPLing system can "chase" the sysplex CDS configuration through the use of residual data in the sysplex CDS's that were previously used by the sysplex. But what can go wrong?

What if the Sysplex CDS chain is broken?



What happens to "chase" if a sysplex CDS is deleted or inaccessible after the PSWITCH?

```
IXC2681 "CDS in COUPLExx not consistent"
IXC275I "Here are CDS defined in COUPLExx"
IXC273I "Looking for CDS last used by sysplex"
IXC275I "Here are CDS last used by sysplex"
IXC222D REPLY U TO USE RESOLVED DATA SETS OR R TO RESPECIFY
        COUPLEXX
```

I don't think I have all the possible operator prompt cases shown here

Found a sysplex, but issues along the way

```
IXC2681 "CDS in COUPLExx not consistent"
IXC275I "Here are CDS defined in COUPLExx"
IXC221D REPLY C TO USE COUPLE DATA SETS
        SPECIFIED IN COUPLExx OR R TO RESPECIFY COUPLEXX
```

Did not find a sysplex

We reconsider the same sequence of sysplex couple data set (CDS) changes as on the previous slide, except the CDS removed from service by the PSWITCH is immediately deleted.

Case 1: Sysplex using P=CDS1, A=CDS2
System joins sysplex with (P=CDS1,A=CDS2) as before since no changes have occurred.

Case 2: Sysplex using P=CDS2, no alternate, CDS1 is not accessible
The IPLing system is unable to open CDS1. It opens CDS2 and finds a self-consistent CDS configuration of (P=CDS2,A=none). Might be the right sysplex. Prompts operator with IXC222D.

Case 3: Sysplex using P=CDS2, A=CDS3, CDS1 is not accessible.
The IPLing system is unable to open CDS1. It opens CDS2 to see (P=CDS2,A=CDS3). It opens CDS3 to see the same CDS configuration in use. Might be the right sysplex. Prompts operator with IXC222D.

Case 4: Sysplex using P=CDS3, no alternate, neither CDS1 nor CDS2 is accessible.
Case 5: Sysplex using P=CDS3, A=CDS4, neither CDS1 nor CDS2 is accessible.
Case 6: Sysplex using P=CDS4, no alternate, neither CDS1 nor CDS2 is accessible.
These cases are identical. The IPLing system cannot open CDS1 and cannot open CDS2. Thus it cannot find the current sysplex. To successfully IPL the system into the sysplex, the COUPLExx parmlib member would have to identify the correct CDS configuration for the current sysplex. Operator is prompted for new COUPLExx

Let us reconsider cases 4, 5, and 6 if CDS1 and CDS3 are deleted after the PSWITCH, but not CDS2.

Case 4: Sysplex using P=CDS3, no alternate, CDS1 not accessible.
The IPLing system cannot open CDS1, it opens CDS2 and sees (P=CDS2,A=CDS3). So it opens CDS3 and sees (P=CDS3,A=none). Might be the right sysplex. Prompts operator with IXC222D.

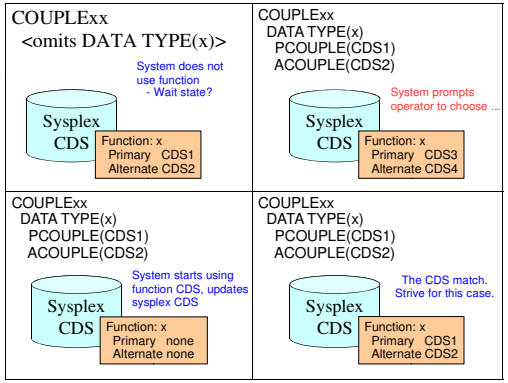
Case 5: Sysplex using P=CDS3, A=CDS4, CDS1 not accessible.
The IPLing system cannot open CDS1, it opens CDS2 and sees (P=CDS2,A=CDS3). So it opens CDS3 and sees (P=CDS3,A=CDS4). It opens CDS4 and sees (P=CDS3,A=CDS4). Prompts operator with IXC222D.

Case 6: Sysplex using P=CDS4, no alternate, neither CDS1 nor CDS3 is accessible.
The IPLing system cannot open CDS1. It opens CDS2 and sees (P=CDS2,A=CDS3). It is unable to open CDS3. Thus it cannot find the current sysplex. To successfully IPL the system into the sysplex, the COUPLExx parmlib member would have to identify (P=CDS4,A=none) as the CDS configuration.

IPLing After Function CDS Configuration Changes

Row applies at sysplex IPL.

Otherwise IPLing system will try to use function CDS described in sysplex CDS.



System Prompts Operator to Choose Function CDS

- Recall that sysplex CDS contains function CDS configuration in use by sysplex
- If COUPLExx DATA statement for a function designates a CDS configuration that does not match the one in the sysplex CDS, the operator is prompted to resolve the discrepancy

```
IXC288I "Here are function CDS defined in COUPLExx"
IXC288I "Here are function CDS last used by sysplex"
IXC289D REPLY U TO USE THE DATA SETS LAST USED FOR function
 OR C TO USE COUPLE DATA SETS SPECIFIED IN COUPLExx
```

Another Possibility

- IPLing system tries to bring CDS into service but:
  - CDS has been previously used by some sysplex
  - The sysplex ownership information in the CDS does not match this sysplex
  - So the operator is prompted:

```
IXC248E "This CDS might be in use by some other sysplex"
IXC247D REPLY U TO ACCEPT USE OR D TO DENY USE
OF THE COUPLE DATA SET FOR function
```

- Accepting use OK if CDS is really meant to be used by this sysplex
  - Perhaps COUPLExx did not match the last used CDS configuration, and
  - At sysplex IPL, operator replied 'C' to IXC289D to use COUPLExx configuration
- But the message is issued because the CDS could be in use by another sysplex. If so, accepting use will rip the CDS away from the other sysplex – and possibly cause an outage
- Do not automate IXC247D with reply of 'U'

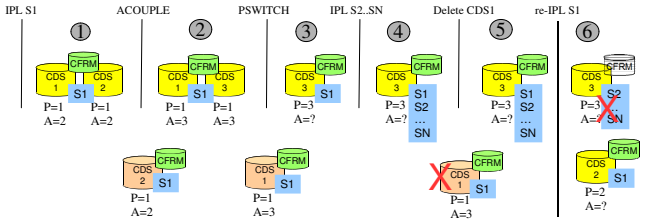
CDS Configuration Discrepancies Implications

- Extra messages during IPL
- Elongates the IPL
- May require operator intervention
  - Will operator make the right decision?
- Generally increases risk of outages ...

If COUPLExx parmlib member COUPLE statement does not identify the Sysplex Couple Data Set (CDS) configuration that is in use by the sysplex, the time to initialize XCF will take longer. In some tests the IPL was observed to increase by around a minute. Clearly the time could vary. If you are at all interested in keeping IPL times as short as possible, you should ensure that the COUPLExx parmlib member accurately reflects the current CDS configuration.

Thus, whenever you switch any of the Couple Data Sets you should always update the COUPLExx member to accurately reflect the current topology, even if the switch is only temporary.

Sysplex CDS Chase Could Lead to System Outage !



For each IPL, COUPLExx indicates:
PCOUPLE(CDS1)
ACOUPLE(CDS2)

- S2..SN chase finds CDS3 since CDS1 exists
- When S1 re-IPLs, CDS1 is gone
  - Residual CDS2 shows S1 as only system in sysplex
  - S1 thinks sysplex re-IPL, starts new sysplex
  - Claims ownership of CFRM CDS
  - S2..SN wait-state when CFRM CDS ownership lost

Behavior prior to z/OS V1R10

Scenario:
In all cases COUPLExx specifies primary CDS1 and alternate CDS2

1) System S1 IPLs into sysplex. Both CDS show S1 as active in the sysplex, using CFRM CDS.

2) System S1 initiates ACOUPLE to bring CDS3 as the new alternate CDS. The old alternate CDS2 is no longer in service but has residual data showing S1 to be the only system in the sysplex with a sysplex CDS configuration of primary CDS1 and alternate CDS2, using CFRM CDS.

3) System S1 initiates PSWITCH to make CDS3 be the primary sysplex CDS. The old primary CDS1 is no longer in service, but has residual data showing a sysplex CDS configuration of primary CDS1 and alternate CDS3.

4) System S2 (through SN) IPL. Looking inside the CDS specified by COUPLExx, the CDS configurations are not consistent. CDS1 shows (P=CDS1,A=CDS3) and CDS2 shows (P=CDS1,A=CDS2). The IPLing system “chases” the sysplex by opening CDS3, which shows (P=CDS3,A=none) which is a self-consistent CDS configuration that could be validly in use by a sysplex. Seeing active systems in CDS3, the IPLing system joins the sysplex.

5) CDS1 is deleted, breaking the chain of CDS that would allow chase processing to find the sysplex.

6) System S1 is reIPLed (in the same LPAR). The COUPLExx parmlib member still identifies CDS1 and CDS2. S1 cannot access CDS1 because the data set was deleted. S1 attempts to continue using only CDS2. However, CDS2 was removed from service before S2..SN joined the sysplex, and so has no record of any other active system. XCF initialization would then treat this as a sysplex IPL, effectively starting a separate sysplex. The IPLing system would establish new ownership information in the function CDS, which has the effect of "stealing" them from the active systems. If the function CDS in use are the same ones described by the sysplex CDS, and still have the same ownership information described by the sysplex CDS, this “stealing” would not result in an IXC248E / IXC247D prompt to ask the operator if stealing was OK. All active systems in the original sysplex would lose all function CDS, which would trigger WAIT 0A2 / 9C on every system where CFRM was in use.

Conclusions:

•Keep the COUPLExx parmlib up to date with the current CDS configuration

•Do not delete a residual sysplex CDS until the COUPLExx parmlib members used to IPL the systems in the sysplex are updated to no longer reference the residual CDS

Might Operator Prompt Save the Day ?

- For IPL of S1 at step 6
  - COUPLExx indicated (P=CDS1,A=CDS2)
  - Chase lands in residual CDS2, which also indicates (P=CDS1,A=CDS2)
  - But CDS1 is not accessible
  - Discrepancy with trouble along the way implies ask
- Operator is prompted with messages that amount to:
  - "COUPLExx said (P=CDS1,A=CDS2)" *Behavior as of z/OS V1R10*
  - "Cannot use CDS1"
  - "Configuration of (P=CDS2,A=none) can be used"
  - **IXC222D REPLY U TO USE RESOLVED DATA SETS OR R TO RESPECIFY COUPLEXX**
- No sysplex is currently using CDS2. The sysplex we want to join is using (P=CDS3,A=none). CDS3 is not mentioned in messages.
- Will the operator recognize the danger?
  - Needs to reIPL with COUPLExx specifying (P=CDS3,A=none)

Does such a COUPLExx even exist?

The previous slide is the behavior that existed prior to z/OS V1R10. In particular, a system would revert to the COUPLExx CDS configuration without prompting the operator if chase processing was unable to resolve the last-used sysplex CDS configuration. As shown by the previous slide, this behavior could lead to a sysplex outage.

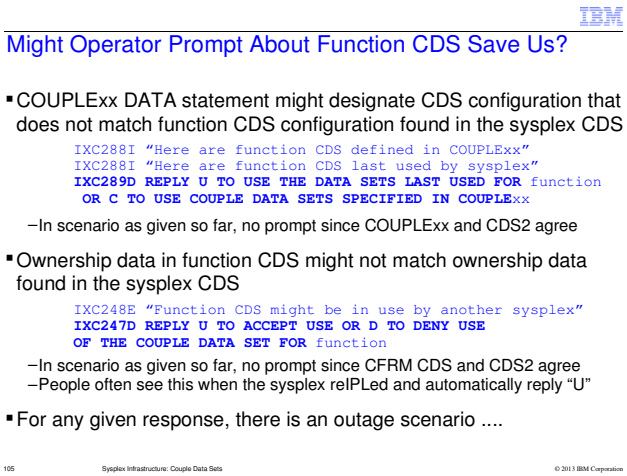
As of z/OS V1R10, various sysplex CDS "discrepancies" encountered during the IPL cause the operator to be prompted to choose a course of action. The specific messages will vary according to the specific situation. In general, the system will issue messages (such as IXC244E, IXC268I, IXC270I, IXC272I, IXC273I, IXC275I) to indicate one or more of the following:

-CDS cannot be used
-CDS specified in COUPLExx are inconsistent (don't have same CDS configuration)
-Going to try to use CDS specified in COUPLExx (chase failed)
-Could not resolve the CDS (chase failed)
-Trying to find CDS last used by sysplex (chasing)
-Report CDS specified in the COUPLExx parmlib member used for the IPL
-Report CDS configuration arrived at via chase processing (CDS last used by sysplex)

The operator might then be prompted (IXC221D, IXC222D, IXC269D) to choose between one or more of the following actions:

-Continue IPL using CDS in COUPLExx
-Continue IPL using CDS resolved by chase processing
-Continue IPL using a suggested CDS configuration
-Start over with different COUPLExx parmlib member

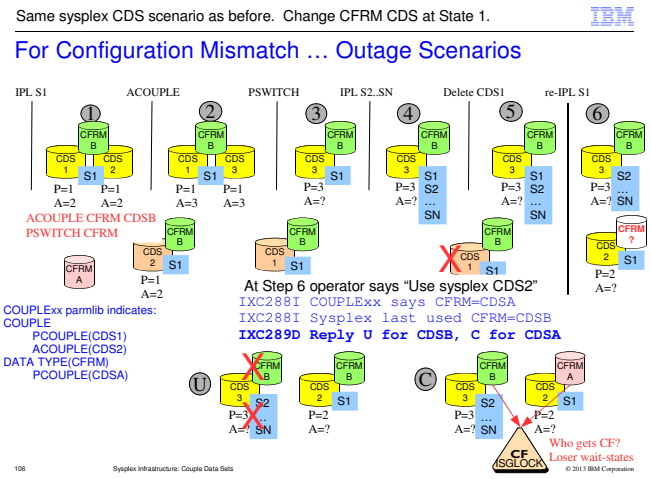
However, I think it is difficult for the operator to determine the correct course of action. In the case described by this slide, none of the CDS listed in the messages are being used by the active sysplex that S1 is trying to join. I think it would take an exceptional operator to recognize that the correct course of action is to start over with a different COUPLExx parmlib member (which might need to be created). Going forward with the proposed CDS configuration of (P=CDS2,A=none) will cause a sysplex wide outage for the active sysplex using (P=CDS3,A=none). The wrong choice is costly indeed! Do not do this to your operator. Keep COUPLExx current with CDS configuration.



Another possible set of messages and operator prompts could be issued regarding the function CDS. The scenario as described on the previous slides concerned the sysplex CDS. After the issues (if any) with the sysplex CDS are resolved, the system will try to gain access to the function CDS. This slide gives some examples of prompts related to the function CDS. These notes describe scenarios where each possible answer leads to a serious outage (such as loss of a system o

So let us continue the scenario from the previous slides. For simplicity I will assume that there is one CFRM CDS in used, call it CDSA and that the COUPLExx parmlib member has a DATA statement identifying CDSA as the primary CDS for CFRM (and shamefully, no alternate). We are again at step 6 and the operator (mistakenly) indicated that system S1 should proceed with using sysplex CDS2. Under the assumption that the same CFRM CDSA has been in use throughout the scenario, system S1 claims ownership of the CFRM CDSA (as was the case prior to z/OS V1R10). That happens because the ownership information in the residual sysplex CDS2 still matches the ownership information in the CFRM CDSA (as well as the ownership information in the sysplex CDS3). Since everything matches, there is no reason to prompt the operator about the function CDS, and systems S2..SN wait-state when S1 claims ownership of CDSA.

But suppose back at step 1 of our scenario, there had been an ACOUPLE of CFRM CDSB followed by a PSWITCH to CDSB. The residual sysplex CDS2 would then show a CFRM CDS configuration of (P=CDSB,A=none). We again arrive at step 6, the operator mistakenly tells S1 to used sysplex CDS2. S1 again thinks it is reIPLing the sysplex. But the CFRM CDS configuration of (P=CDSA,A=none) in the COUPLExx does not match the CFRM CDS configuration last used by the sysplex according to sysplex CDS2, namely (P=CDSB,A=none). Since there is a discrepancy, S1 prompts the operator (first bullet), in effect asking "Should I use CFRM CDSA or CDSB?" The first bullet illustrates the case where a system is IPLed with COUPLExx parmlib member in which the function CDS configuration (for a particular) function does not match the function CDS configuration last used by the sysplex. For example, the sysplex

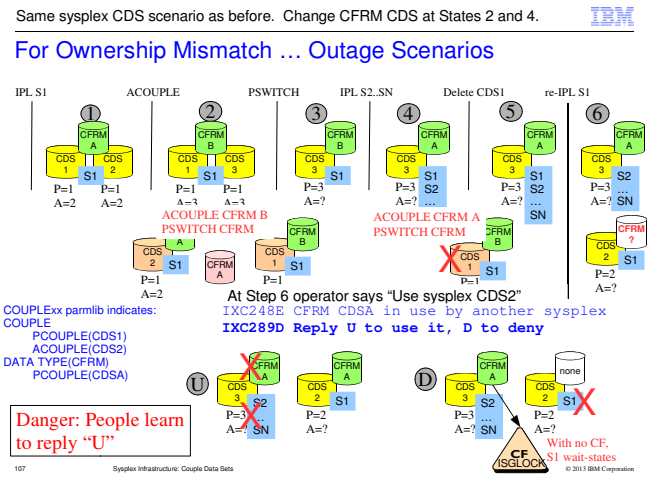


Run the same scenario with respect to the sysplex CDS configuration changes. However, after system S1 IPLs at step 1, the operator issues SETXCF commands to bring CDSB into service as the primary CDS for the CFRM function. CFRM CDSA is removed from service but has residual information. In particular, both CDSA and CDSB have the same policies. As CDSB is used, the dynamic content of the active policy will be updated and diverge from the residual data in CDSA. However, the key point for our scenario is that the active policy in both CDS defines the same set of coupling facilities.

Now continue the sysplex CDS scenario. When S1 is reIPLed at step 6, the operator is prompted and (mistakenly) indicates to continue the IPL with sysplex CDS2. S1 brings CDS2 into service successfully and then looks to bring the function CDS into service. In this example, we use the CFRM function for illustration. The principles are the same for the other functions, but the specific results would differ (and may or may not be equally catastrophic). At any rate, since S1 believes it is starting a new sysplex, it compares the CFRM CDS configuration identified by COUPLExx parmlib member to the CFRM CDS configuration recorded in sysplex CDS2. Since (P=CDSA,A=none) in COUPLExx does not match (P=CDSB,A=none) recorded in CDS2, the operator is prompted to decide which configuration should be used.

If the operator replies "U" to use the configuration last used by the sysplex, system S1 will claim ownership of CDSB. But CDSB is currently in use by systems S2...SN. When those systems recognize that they no longer have ownership of CDSB, they wait-state due to loss of the CFRM CDS.

If the operator instead replies "C" to use the configuration identified in the COUPLExx parmlib member, S1 will claim ownership of CDSA. So S1 is using CDSA and S2..SN are using CDSB. However, CDSA has an active policy that identifies the same coupling facilities as the active policy in CDSB. System S1 will try to claim ownership of those coupling facilities. Since they are in use by a different sysplex (S2..SN), the operator is prompted. If the operator tells S1 to proceed with claiming ownership, systems S2..SN lose their coupling facilities. If the operator denies the use, system S1 will not have access to any coupling facilities. Just to make it really painful, I assume GRS Star-Mode. With no access to the ISGLOCK structure, a system is wait-stated. Even if the system survives without the CF, it will likely be severely impaired.



Run the same scenario with respect to the sysplex CDS configuration changes. However, after system S1 ACOUPLEs in sysplex CDS3 at step 2, the operator issues SETXCF commands to bring CDSB into service as the primary CDS for the CFRM function. CFRM CDSA is removed from service. Later, after systems S2..SN are IPLed into the sysplex at step 4, the operator issues SETXCF commands to bring CDSA back into service as the primary CDS for the CFRM function. As part of bringing CDSA into service, it will be updated with new ownership information. For this example, the key point is that the residual information in sysplex CDS2 has the original CFRM CDSA ownership information which differs from the ownership information recorded in CDSA from when it was brought into service the second time.

Now continue the sysplex CDS scenario. When S1 is reIPLed at step 6, the operator is prompted and (mistakenly) indicates to continue the IPL with sysplex CDS2. S1 brings CDS2 into service successfully and then looks to bring the function CDS into service. Since S1 believes it is starting a new sysplex, it compares the CFRM CDS configuration identified by COUPLExx parmlib member to the CFRM CDS configuration recorded in sysplex CDS2. In this case they match: both have (P=CDSA,A=none). So system S1 will attempt to claim ownership of CDSA. But the ownership information in CDSA does not match the ownership information recorded in sysplex CDS2. This could imply that CDSA is in use by another sysplex. So S1 prompts the operator.

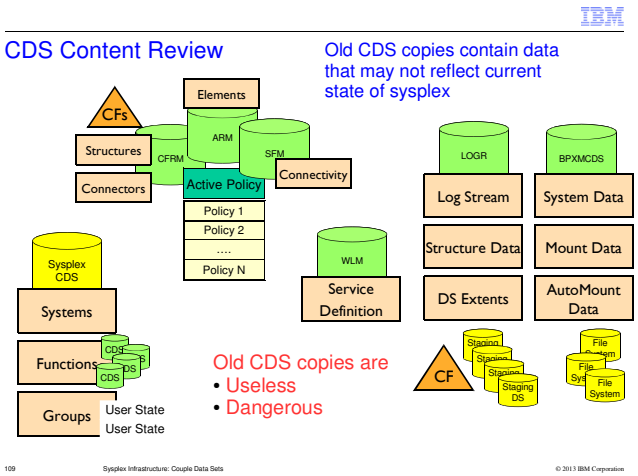
If the operator replies "U" to proceed with claiming CFRM CDSA, S1 updates the ownership information in CDSA. When systems S2...SN recognize that they no longer have ownership, they wait-state due to loss of the CFRM CDS.

If the operator instead replies "D" to deny use of CDSA, system S1 will not have access to any coupling facilities. Just to make it really painful, I assume GRS Star-Mode. With no access to the ISGLOCK structure, a system is wait-stated.

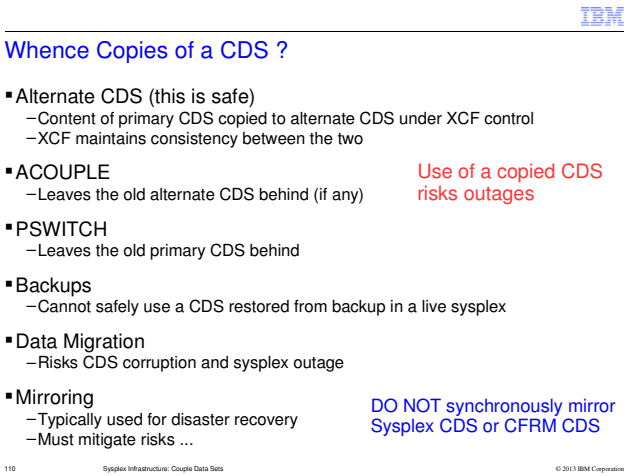
Danger: When doing a sysplex-wide reIPL, "U" is generally the correct response. For installations that regularly reIPL the sysplex, the operators learn to automatically make that reply because it is "always" right. Some installations even automate the response. As suggested here, there are scenarios where "U" can induce outages. This is not the only scenario. It really takes careful consideration and operator awareness to get the correct response. If the operator fails to reply, the safest automated response is "D" to deny use.

What Went Wrong? … Three perspectives

- Failed to update COUPLExx to reflect current CDS configuration
  - Or operator IPLed with the wrong COUPLExx
- Prematurely deleted CDS
  - CDS should not be deleted until COUPLExx is updated
  - But impact is same if CDS still exists but is not accessible
    - Which may not be avoidable (so remains a concern)
- Used stale copy of CDS
  - In general, incorrect use of an old CDS can lead to trouble
  - Previous slide shows it can be an issue when IPLing into an existing sysplex
  - More typically it is an issue when IPLing a sysplex
  - So let us consider the issue of CDS copies in more detail ...



The CDS contain state information that relates to a specific instance of a specific sysplex. You must assume that any CDS that is not the current CDS in use by the sysplex has stale information that does not reflect the current state of the sysplex. IPLing a sysplex with stale CDS containing old state information can lead to all sorts of confusion, failures, and outages. The previous slides have tried to make that point. Installations have proven this assertion time and time again. We know because we see the resulting PMRs.



ACOUPLE

PSWITCH

DASD Backup (or an explicit copy)

Data Migration

What drives data migration? Generally a technology refresh of your storage equipment. You might need to move all the data from one unit to another when:●The lease on a piece of storage equipment ends.●Data is being moved from old to new storage.●The storage system is being upgraded.●The organization has decided to standardize on a particular technology.

Mirroring

Avoid synchronous mirroring of sysplex couple data sets because sysplex performance is impeded when you access CDSs that are mirrored synchronously. XCF maintains the primary and alternate CDSs. The write to the CDS is not complete until the local copy and the remote copy of the CDS are successfully written. The additional overhead of synchronous mirroring can cause delays in applications that need to access the CDS. If a delay occurs or the there are long busy conditions on the mirrored copy, I/O delays are reported for attempts to access CDSs. In some cases, the mirroring technology might be suspended. In other situations, if the delays are long enough, a permanent I/O error condition against the CDS can occur. XCF takes the CDS that encountered the error out of service. If both the primary CDS and the alternate CDS encounter a permanent I/O error, the functions provided by the CDS will cease immediately. If it is the SYSPLEX CDS or the CFRM CDS encountering the error, a sysplex outage occurs.

Using Copies of Couple Data Sets is Risky

- Customer Goal: Simplify DR configuration and minimize time to recovery by copying CDS to DR site
  - Allows all volumes on device to be copied or asynchronously mirrored without need for manual exclusion of volumes containing CDS
  - Need not run format utility to create CDS at DR site
  - Need not run data utility to create policies in the CDS at DR site
- Proven Risks:
  - Sysplex outages
  - Data Integrity issues
  - CDS at primary site unexpectedly removed from service
  - 0A3 wait-state when GRS cannot allocate ISGLOCK structure
  - Residual data for inaccessible structures at other site
  - One, some, or all CF's ripped away from active sysplex

Copies defeat protection mechanisms.
Hard to anticipate ways things can go wrong.

- Warning: Requires great care to avoid sysplex outages

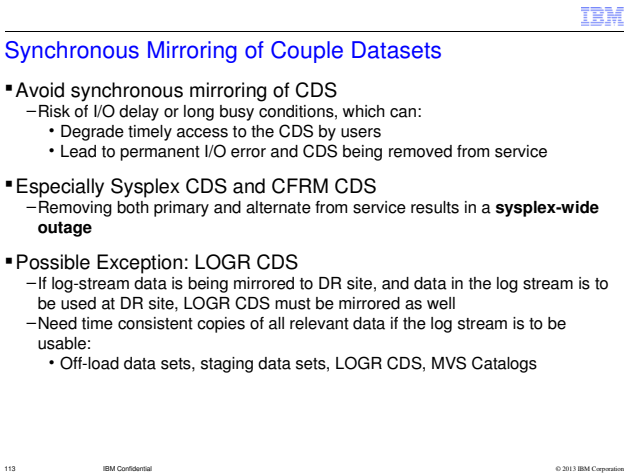
Software mirroring, always have a primary and alternate CDS is a MUST!!

Hardware level mirroring includes PPRC, XCR, metro mirror, global mirror, etc.

If Using Copies of Couple Data Sets at DR Site

Applies to any test environment that uses a copy of a CDS

- All CFs used by the DR site should be defined in the CFRM policy used by the primary site
- Never allow DR site to gain access to CF's in primary site
- Never allow DR site to gain access to DASD in primary site
- When configuration changes, be sure you maintain these conditions
  - Needs to be part of your change management procedures
- Reference: Hot Topics February 2011 Issue 24 p.69 "*Mirror, mirror, on the wall, should couple dataset be mirrored at all?*"
  - *http://publibfp.dhe.ibm.com/epubs/pdf/eoz2n1c0.pdf*
- Reference: WP102281 "Couple Data Sets: Best Practices and Avoiding Disasters"
  - *http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102281*



Avoid synchronous mirroring of sysplex couple data sets because sysplex performance is impeded when you access CDSs that are mirrored synchronously. XCF maintains the primary and alternate CDSs. The write to the CDS is not complete until the local copy and the remote copy of the CDS are successfully written. The additional overhead of synchronous mirroring can cause delays in applications that need to access the CDS. If a delay occurs or the there are long busy conditions on the mirrored copy, I/O delays are reported for attempts to access CDSs. In some cases, the mirroring technology might be suspended. In other situations, if the delays are long enough, a permanent I/O error condition against the CDS can occur. XCF takes the CDS that encountered the error out of service. If both the primary CDS and the alternate CDS encounter a permanent I/O error, the functions provided by the CDS will cease immediately. If it is the SYSPLEX CDS or the CFRM CDS encountering the error, a sysplex outage occurs.

Asynchronous mirroring … Recently Discovered Issue

- Asynchronously mirrored CDS may not be usable
  - The control unit copies data on a track basis, without regard for grouping of tracks within logical records.
  - So an asynchronously-maintained copy of a CDS at a DR site might, at any given point in time, have an internally inconsistent set of tracks within a record.
  - If you fail over to the DR site at such a time, the plex won't be able to use that CDS.
- Not yet solved
  - So stage freshly-formatted CDSes at each potential fail-over site instead of using mirrored CDSes

Possible Approach To Handling DR and Mirroring Issues

- Create and mirror separate CDS for use at DR site
  - These CDS are never used by production sysplex
  - Whenever you format CDS for production, format one for DR
  - Whenever you update administrative policies, make a parallel update in the CDS to be used at DR site
    - For CFRM, there will be differences due to CF identification
- If need mirrored copy of LOGR CDS
  - Do not mirror any other production CDS
  - Same sysplex name for DR and production
  - Maintain COUPLExx for use at DR site that points at:
    - DR copy of the mirrored LOGR CDS that is part of the consistency group
    - DR copy of the other mirrored CDS created for the DR site

If GDPS, use their procedures

- If don't need mirrored LOGR CDS
  - Do not mirror any production CDS
  - Use different sysplex name for DR and production
  - Maintain COUPLExx for use at DR site that points at:
    - DR copy of all mirrored CDS created for the DR site

What CDS when IPLing System ?

- When IPLing system into existing sysplex
  - Best: COUPLExx identifies CDS currently in use by sysplex
  - Maybe: COUPLExx identifies Sysplex CDS allowing “chase”
  - Bad: COUPLExx identifies stale copy of CDS
- When re-IPLing the sysplex:
  - Best: IPL with CDS most recently in use by the sysplex
  - OK: IPL with previously unused CDS (freshly formatted)
  - Bad: IPL with stale copy of CDS
- When IPLing sysplex at DR site
  - Best: IPL with previously unused CDS (freshly formatted)
    Except perhaps LOGR CDS that is part of consistency group
  - Risky: Anything else
    Understand the risks and mitigate/eliminate them

ReIPL of Sysplex and CFRM Policy

- Persistence of active CFRM policy can be a problem if you need to reIPL the sysplex with a different policy
- ReIPL of sysplex suggests some catastrophe
  - Or a significant reconfiguration of the sysplex
- Need for a new policy suggests the sysplex won't come back up with the policy that was last in use by the sysplex
  - Hardware issues
  - Errors in policy
  - Production policy not suitable for use at DR site (wrong CF's)
- Which suggests people are upset
  - Why are we still down?
  - Why won't the new machines come up?
- So how to restart sysplex with a different policy?

Restarting Sysplex with Different CFRM Policy

- See FLASH10786: *Where is my coupling facility?*
  - First system cannot IPL into sysplex since “no” CF for ISGLOCK structure
  - Nice summary of common errors when updating policies
  - Describes technique to IPL sysplex with a repaired (new) policy
  - Probably the most reasonable and least disruptive way forward
- Next safest alternative is newly formatted CDS (all of them)
  - Lose all state information, but everything will be self consistent
  - Will need to recreate and start policies
  - Time to recover data bases and transactions could be an issue if prior instance of sysplex did not shut down normally
- Fail over to DR site ?
- Possibly other options, but likely quite risky
  - Potential for inconsistencies that will compound troubles
  - Depends on the exploiters and their implementation
  - Not clear that a component expert can reliably assess situation for you

www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10786

Abstract: The first system to IPL in a sysplex in GRS STAR mode is unable to allocate the ISGLOCK structure because no coupling facility is accessible. This paper provides clear and concise instructions on how to address this issue.

Use SETXCF command to stop the current active policy for a CFRM CDS that is currently in use by the sysplex

Implies all CF structures deleted (not possible if GRS-Star mode)

My Conclusions

- CDS Placement
  - Dedicated volume
  - Not to be backed up, migrated, or mirrored
- Do not use copies of CDS
  - Increases complexity
  - Understand risks and eliminate them
- Manage a group of CDS
  - Format and maintain primary, alternate, and spares as a unit
  - Delete the set when no longer needed
- Keep COUPLExx in sync with CDS configuration
- Need to be very careful and certain when responding to XCF prompts during IPL to avoid outages
  - Use good naming conventions to help avoid confusion
  - Maintain accurate configuration information

Dumping CDS Content

- IBM Service Personnel may need to see internal content of a CDS if there is an issue or suspected problem
- Use ADRDSSU utility to dump a CDS

```
//DUMP JOB MSGLEVEL=(1,1)
//STEP1 EXEC PGM=ADRDSSU,REGION=4M
//SYSPRINT DD SYSOUT=*
//DD1 DD DISP=SHR,VOL=SER=volser,UNIT=unit
//SYSIN DD *
PRINT DATASET(cds name) INDDNAME(DD1)TOL(ENQF)
/*
```

Only needed if dumping in-use CDS

XCF Health Checks for CDS

- XCF_CDS_MAXSYSTEM
  - MAXSYSTEM of function CDS should be >= MAXSYSTEM of sysplex CDS
- XCF_CDS_SEPARATION
  - Performance sensitive CDS's should not be on same volume
- XCF_CDS_SPOF
  - Primary and alternate CDS should be failure isolated
- XCF_SYSPLEX_CDS_CAPACITY
  - Does the sysplex CDS appear to be formatted with sufficient capacity to allow growth

Some functions might also have their own CDS related health checks

XCF_CDS_MAXSYSTEM
It is recommended that each couple data set defined to the sysplex be formatted with a MAXSYSTEM value that is at least equal to the value defined in the primary sysplex CDS. If a function CDS has a smaller MAXSYSTEM value, then asystem joining the sysplex with a higher slot number would not be able to use the function provided by that function CDS.

XCF_CDS_SEPARATION
Check that performance-sensitive couple data sets are isolated from each other. The sysplex and CFRM primary couple data sets reside on different volumes. The LOGR primary CDS reside on a volume separate from the sysplex and CFRM primaries if in the installation's view the rate of I/O activity to the LOGR CDS warrants it

XCF_CDS_SPOF
Check that couple data sets are configured without single pointsof failure.
Each primary couple data set should reside on a different volume than its corresponding alternate couple data set.

The I/O configuration not create any "hidden" single points of failure, e.g., placing the volumes containing both primary and alternate of a given type in a single physical device or behind the same switch.

XCF_SYSPLEX_CDS_CAPACITY
Check that the maximum number of systems, groups, and members have not at some time reached a threshold determined by the best practice amount of space required for growth of systems, groups, and members.

Summary

- Sysplex Couple Data Set
- Function Couple Data Set
- Format Utility
- Data Utility
- Primary CDS
- Alternate CDS
- PSWITCH
- ACOUPLE
- Version or format level
- Defining CDS to sysplex
- Defining policies
- Changing CDS configuration
- Maintaining COUPLExx
- Perils of CDS copies
- Caveats for Test
- Dumping CDS
- CDS Health checks

For More Information

- z/OS Publications
  - z/OS MVS Setting Up A Sysplex
  - z/OS UNIX System Services Planning
  - z/OS MVS Planning: Global Resource Serialization
  - z/OS MVS Planning: Workload Management
- Redbooks
  - System z Parallel Sysplex Best Practices
  - System z Programmer's Guide to: z/OS System Logger
- Learn from the misfortune of others
  - WP102281 Couple Data Sets: Best Practices and Avoiding Disasters
  - FLASH10786: Where is my coupling facility?
  - Hot Topics Article: Mirror, mirror on the wall, should these data sets be mirrored at all?

z/OS Publications are available at $$$

Redbooks are availale at $$$

White Papers are available ag

document can be found on the web, www.ibm.com/support/techdocs

Under the category of "White Papers."

Hot Topics Article: http://publibfp.dhe.ibm.com/epubs/pdf/eoz2n1c0.pdf

Thank You

Please complete your session evaluation at
SHARE.org/BostonEval

Session 14229
Sysplex Infrastructure:
The Care and Feeding of Couple Data Sets