Parallel Sysplex: Operations, Troubleshooting & Recovery

 MPOR

Parallel Sysplex: Operations, Troubleshooting & Recovery

Contents *Page*

Parallel Sysplex: Operations, Troubleshooting & Recovery

Contents *Page*

Parallel Sysplex: Operations, Troubleshooting & Recovery

Contents *Page*

Parallel Sysplex: Operations, Troubleshooting & Recovery

Contents *Page*

Chapter 1
Runtime Problem Determination

1.1 Segment overview

OK, OK, so you never get any of those problems at IPL time.

Doesn't mean your luck's going to last forever, there's plenty of time for things to go wrong yet . . .

Objectives

On completing this segment of the course, you will be able to:

- **identify the different types of sysplex-related error conditions**
- **deal with the connectivity problems in the sysplex**
- **respond correctly to 'status update missing' conditions**
- **manage the Sysplex Failure Manager environment**
- **respond appropriately to sysplex timer related problems**
- **handle Coupling Facility environment errors**
- **recognise and respond to structure-related errors for the major application systems**
- **operate successfully in the Automatic Restart Manager environment.**

MPOR - 05 - 2

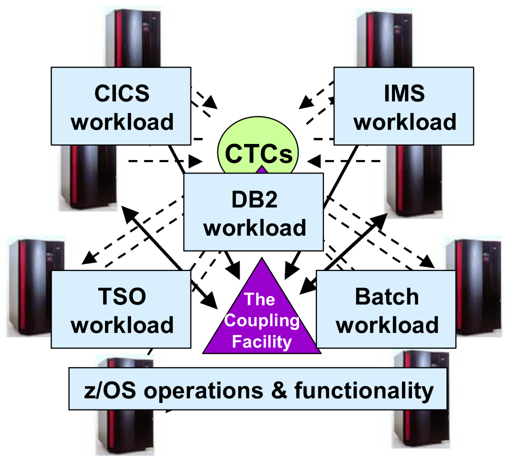
1.2 It's the sysplex that counts . . .

As we said in the introduction, a parallel sysplex can get rather large, consisting of up to 32 interconnected z/OS systems. Even so, through the use of the various signalling and data sharing services, it can still present a single-image environment for both the installation and the workload.

A significant part of this process is the ability to provide an ongoing stable environment for the workload, regardless of outages in individual hardware or software components.

If a major component fails, the sysplex, as an entity, can just 'shrug off' the failure, reconfigure, and resume normal operations. If everything is set up correctly, this requires no external intervention.

It's the sysplex that counts...



A parallel sysplex:

- may consist of up to 32 systems
- and can accept new systems up to that limit dynamically
- but can provide a '**single image**' for the workloads
- **can recover failing work units automatically**, anywhere in the sysplex
- **can provide continuous availability** for application workloads

So how do we keep things running?

MPOR - 05 - 3

1.3 . . . not the individual systems

One of the key things to understand about parallel sysplex is that individual z/OS systems are now relegated to the same position as individual jobs in the non-sysplex environment:

- we care about them,
- but we're prepared to sacrifice them if it's for the 'greater good'.

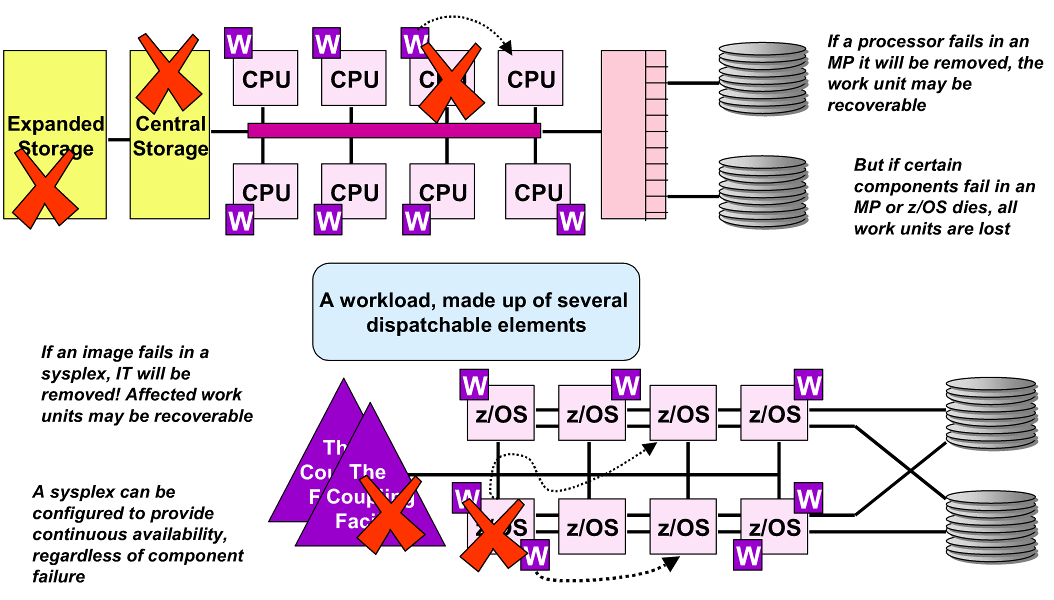
What we mean here is that the failure management policy of z/OS has always been *"contain any damage, remove the source of the damage, but keep the system running".* *z/O*S will remove **any** hardware component or address space that has failed rather than let any damage spread to other components, possibly resulting in the loss of the system.

That policy is now enhanced to enable the **sysplex** to survive any potential damage. What it now means is *"contain any damage in the sysplex, remove the system causing the problem, but keep the sysplex running".*

THE SYSPLEX IS THE IMPORTANT THING,
BECAUSE WORK RUNS "IN THE SYSPLEX"

For the sysplex to be fully functional, all participating systems must be able to communicate fully (i.e. access the signalling functions), and all systems must have access to any shared data in the Coupling Facilities used by those systems. Any system that loses either of these attributes is damaged, and can't participate in the sysplex.

...not the individual systems



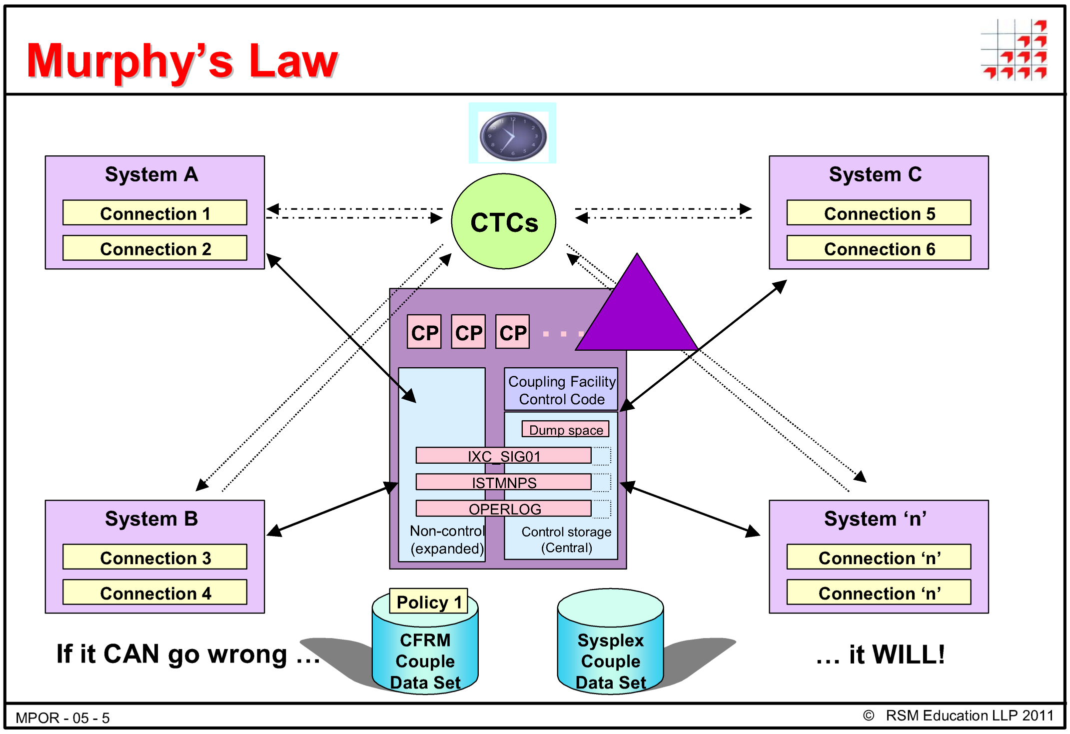
MPOR - 05 - 4

1.4 Murphy's Law

There is a general rule in life (some say) that says "if it can go wrong, it will". Whether you subscribe to this theory or not, it has to be said that we can't allow a failing component in the sysplex to bring the whole sysplex down, any more than we want a failing address space to bring a z/OS system down.

It's quite likely that you won't argue by now if we say that a parallel sysplex is a complex environment.

There are plenty of places where things could break down, with potentially serious results.


Murphy's Law
System A
Connection 1
Connection 2
CTCs
System C
Connection 5
Connection 6
CP
CP
CP
Coupling Facility Control Code
Dump space
IXC_SIG01
ISTMNPS
OPERLOG
Non-control (expanded)
Control storage (Central)
System B
Connection 3
Connection 4
System 'n'
Connection 'n'
Connection 'n'
Policy 1
CFRM Couple Data Set
Sysplex Couple Data Set
If it CAN go wrong …
… it WILL!
MPOR - 05 - 5
© RSM Education LLP 2011

1.5 Redundancy is good for you!

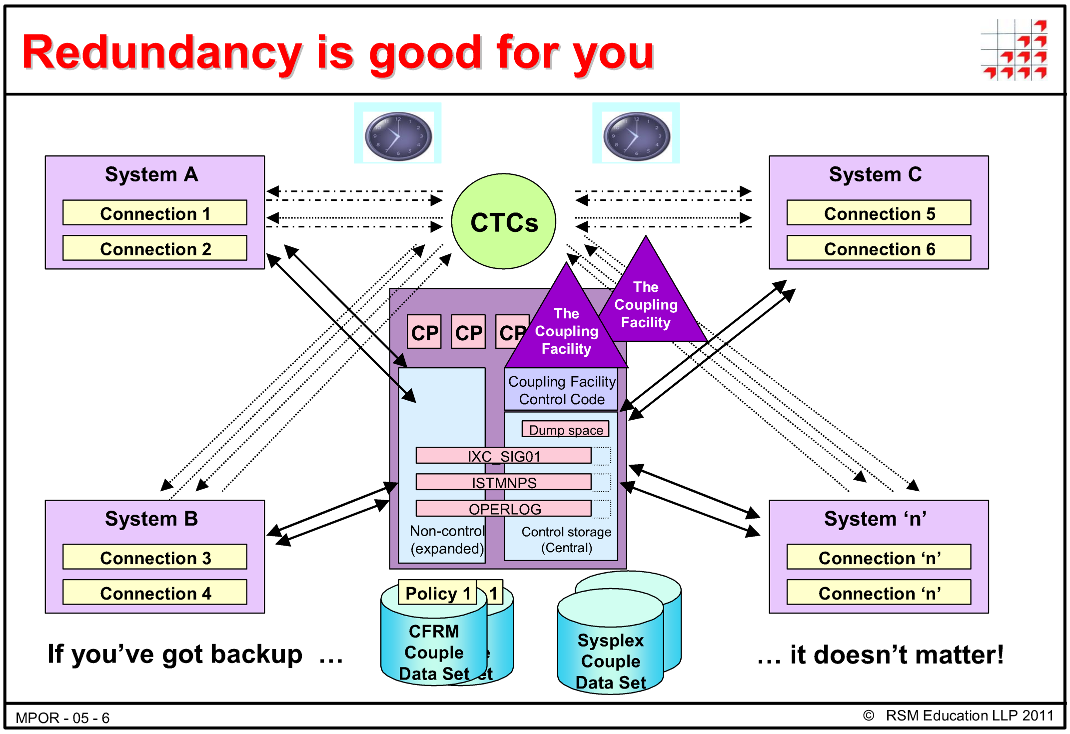
You can help your sysplex's chances of surviving a major component failure by duplicating key components:

- if you need a sysplex timer, use two. They can be cross-configured to act as backup for each other
- double up on your signalling paths, gave two sets of CTCs if your using them for signalling, or have two CF channels from each system if your using them for signalling. Better still have two sets of both and use them all!
- have alternate Couple Data Sets available at all times, and have spares predefined and available as well
- have more than a single Coupling Facility, and make sure you've doubled up on the CF channels for **each** facility

Cost versus security

Yes, this is not the cheapest option, but you have to accept that having a single point of failure anywhere in the sysplex is asking for trouble. In most cases, without redundancy built in, if you lose one of these components you are going to lose **at least one z/OS image,** and in some cases you will lose **the entire sysplex.**

Considering the investment involved in a fully configured sysplex of maybe four or five images, even without duplication, the extra cost of redundancy may well be worth paying.


Redundancy is good for you
System A
Connection 1
Connection 2
CTCs
System C
Connection 5
Connection 6
CP
CP
CP
The Coupling Facility
The Coupling Facility
Coupling Facility Control Code
Dump space
IXC_SIG01
ISTMNPS
OPERLOG
Non-control (expanded)
Control storage (Central)
System B
Connection 3
Connection 4
System 'n'
Connection 'n'
Connection 'n'
Policy 1
CFRM Couple Data Set
Sysplex Couple Data Set
If you've got backup …
… it doesn't matter!
MPOR - 05 - 6
© RSM Education LLP 2011

1.6 Our example configuration

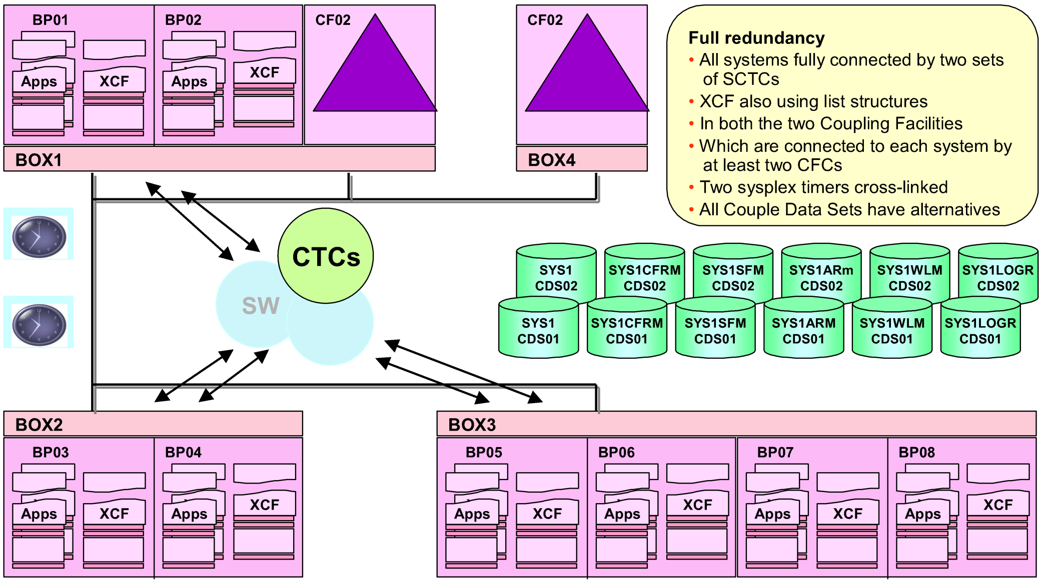
A reminder of our example configuration:

- we are running eight systems in our sysplex
- all systems are fully connected for signalling purposes by two sets of switched CTCs connected via two ESCON switches
- and we're also using two list structures for signalling
- with two Coupling Facilities
- which are in turn both connected to all systems
- and each system connects to each Coupling Facility by at least two Coupling Facility channels
- we have two sysplex timers cross-connected and each machine is connected to both timers
- and all our Couple Data Sets have alternates online, and the primaries and alternates are on RAID devices on separate paths.

So, are we feeling smug or what?

Does our hardware salesman drive a Porsche?

Example configuration



Full redundancy
- All systems fully connected by two sets of SCTCs
- XCF also using list structures
- In both the two Coupling Facilities
- Which are connected to each system by at least two CFCs
- Two sysplex timers cross-linked
- All Couple Data Sets have alternatives

MPOR - 05 - 7

1.7 Failure events and recovery options

Listed opposite are the potential problem areas. Notice that in the applicable areas, where there **IS** redundancy built in to the sysplex, **there is no outage, just a (probably temporary) loss of capacity.**

This part of the course will look at the error situations and how they can be handled, and we'll do so by looking at the six possible options in sequence.

The options are:

1) **Isolating a system due to a physical connectivity problem.** This can be automated by the Sysplex Failure Manager using information provided in an SFM policy.

2) **Isolating a system when a system fails.** This can also be automated by SFM in conjunction with some COUPLEnn parameters.

Both of the above situations can be managed automatically by the **Sysplex Failure Manager** component of XCF.

3) **Dropping a system into a wait state due to ETR failure.** Not much of a recovery option, you might think. And you'd be right.

4) **Dealing with Couple Data Set loss.** One of the great 'it depends' in the recovery environment. All will be revealed later on.

5) **Rebuilding a structure.** This is handled by a combination of SFM action and activities initiated by the affected connections themselves.

6) **Restarting failed applications.** This is handled via the Automatic Restart Manager, which is controlled by an ARM policy. ARM also requires that applications issue certain ARM service macros before they can utilise this facility.

Failure events & recovery options

Failing component	Have backup or alternate	No backup or alternate
XCF path (via CTC)	lose capacity	isolate system ①
XCF path (via List Structures)		
Coupling Facility	rebuild structures	isolate system(s) ①
Coupling Facility Channel failure	lose capacity	isolate system(s) ①
Structure failure (CF ok)	lose capacity	rebuild structures ⑤
MVS system ("status update missing")	n/a	isolate system ②
Sysplex Timer	carry on	wait state ③
Couple Data Set	duplexed pair	wait state ④
Coupling Facility environment	rebuild structures	appl dependent ⑤
Application (non-signalling) structure loss		
Application (batch job or STC)	n/a	invoke ARM ⑥

MPOR - 05 - 8

1.8 CTC signalling path reconfiguration - 1

To start with, let's look at what might happen if you lose a CTC signalling path between two systems, but you've got at least one other path configured and online.

1) In the picture opposite, a **D XCF,PO** is issued on BP01, and you can see that there are two paths going out to BP03. This is just a section of the resulting display.

2) One of those paths experiences a permanent I/O error, and is boxed by z/OS.

3) XCF detects that the path has gone missing, and issues the **IXC467I** and **IXC307I** messages to indicate that the path has now been stopped to XCF.

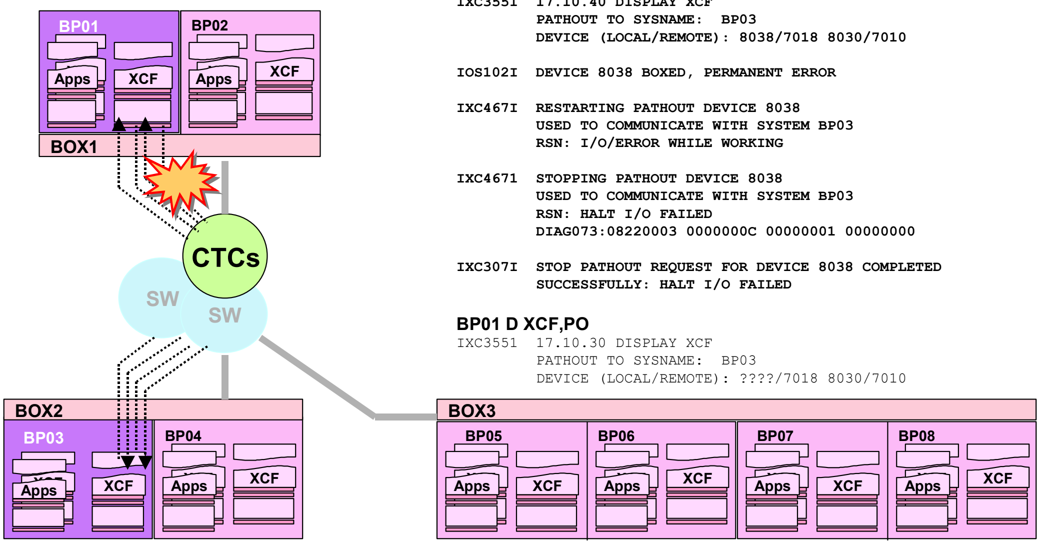
   Note: both these messages have dozens of possible message texts. You may very well see many variations on these messages in the sysplex environment.

4) If we display the paths again, we can see that there is now only one path known to XCF between BP01 and BP03.

(continued on the next two pages)

CTC signalling path reconfiguration - 1

(more than one CTC path available)



BP01 D XCF,PO

```
IXC3551  17.10.40 DISPLAY XCF
         PATHOUT TO SYSNAME:  BP03
         DEVICE (LOCAL/REMOTE): 8038/7018 8030/7010

IOS102I  DEVICE 8038 BOXED, PERMANENT ERROR

IXC467I  RESTARTING PATHOUT DEVICE 8038
         USED TO COMMUNICATE WITH SYSTEM BP03
         RSN: I/O/ERROR WHILE WORKING

IXC4671  STOPPING PATHOUT DEVICE 8038
         USED TO COMMUNICATE WITH SYSTEM BP03
         RSN: HALT I/O FAILED
         DIAG073:08220003 0000000C 00000001 00000000

IXC307I  STOP PATHOUT REQUEST FOR DEVICE 8038 COMPLETED
         SUCCESSFULLY: HALT I/O FAILED
```

BP01 D XCF,PO

```
IXC3551  17.10.30 DISPLAY XCF
         PATHOUT TO SYSNAME:  BP03
         DEVICE (LOCAL/REMOTE): ????/7018 8030/7010
```

MPOR - 05 - 9

1.9 CTC signalling path reconfiguration - 2

(continued from the previous two pages)

1) We start with the display showing the single available path

2) The boxed device is varied online successfully (the **UNCOND** is required to bring a boxed device online)

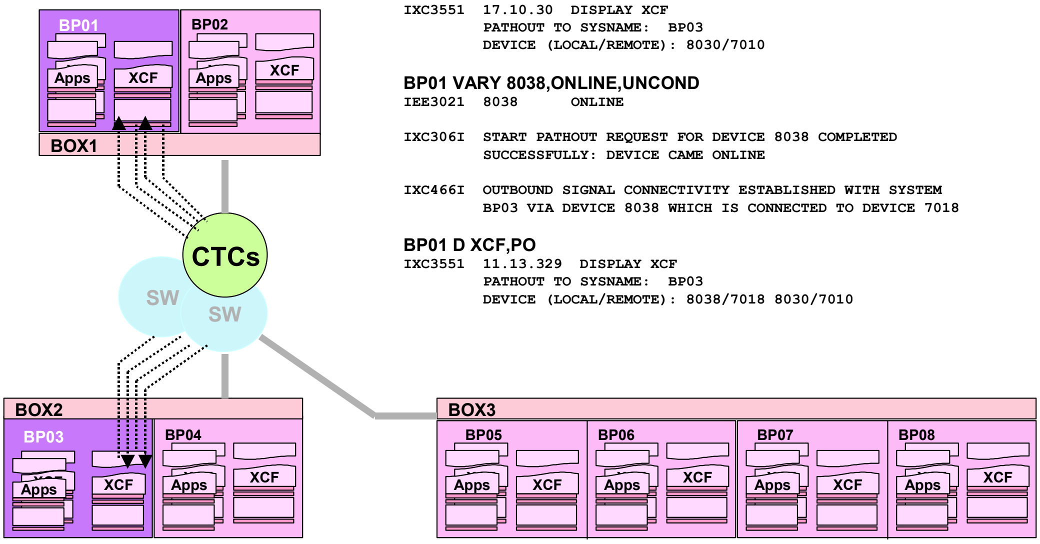
3) XCF detects that its path device is now online again

4) And the final display shows both paths available again

Notice that there was no disruption to sysplex processing while this all took place. Even if we hadn't been able to bring the failed path back into service, sysplex processing would have continued.

CTC signalling path reconfiguration - 2

(more than one CTC path available)



BP01 D XCF,PO

```
IXC3551  17.10.30  DISPLAY XCF
         PATHOUT TO SYSNAME:  BP03
         DEVICE (LOCAL/REMOTE): 8030/7010
```

BP01 VARY 8038,ONLINE,UNCOND

```
IEE3021  8038      ONLINE

IXC306I  START PATHOUT REQUEST FOR DEVICE 8038 COMPLETED
         SUCCESSFULLY: DEVICE CAME ONLINE

IXC466I  OUTBOUND SIGNAL CONNECTIVITY ESTABLISHED WITH SYSTEM
         BP03 VIA DEVICE 8038 WHICH IS CONNECTED TO DEVICE 7018
```

BP01 D XCF,PO

```
IXC3551  11.13.329  DISPLAY XCF
         PATHOUT TO SYSNAME:  BP03
         DEVICE (LOCAL/REMOTE): 8038/7018 8030/7010
```

MPOR - 05 - 10

1.10 Losing the last or only CTC signalling path

Losing the **only** (or last) signalling path between two systems is another matter.

1) The display command shows only one path is available between BP01 and BP03

2) It is stopped by XCF due to an excessive number of retries

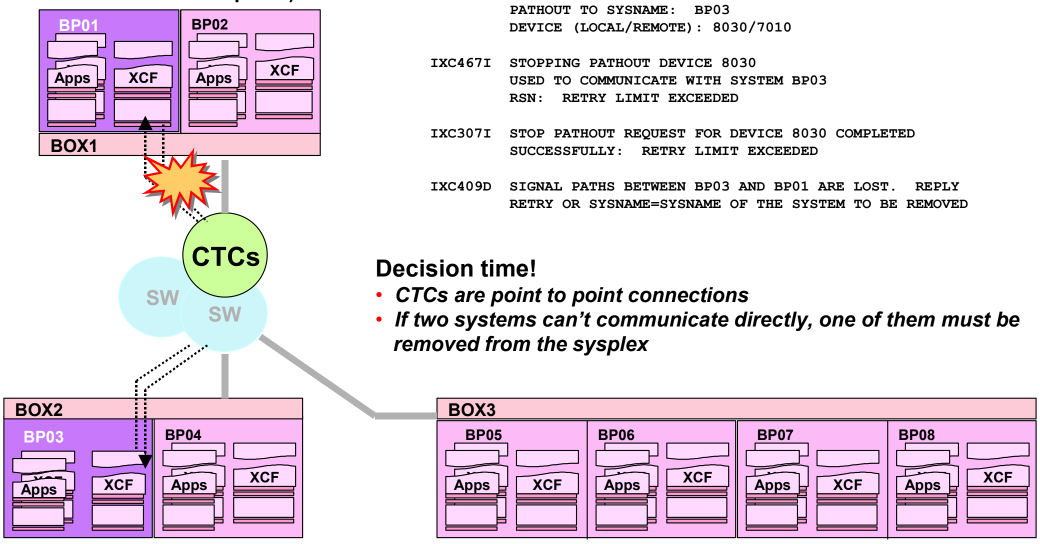
3) Message IXC409D is issued, asking the operators to decide ***which of the two systems should be removed from the sysplex***.

If the only direct connection between two systems in a sysplex is lost, one of the systems must be removed from the sysplex.

We can continue with a perfectly healthy sysplex consisting of seven systems, as both BP01 and BP03 are still capable of communicating directly with everyone else except each other. The question is, which one should stay and which one be removed?

Losing the last or only CTC signalling path

(only one CTC path available
and no structure paths)



BP01 D XCF,PO

```
IXC355I  17.10.40  DISPLAY XCF
         PATHOUT TO SYSNAME:  BP03
         DEVICE (LOCAL/REMOTE): 8030/7010

IXC467I  STOPPING PATHOUT DEVICE 8030
         USED TO COMMUNICATE WITH SYSTEM BP03
         RSN:  RETRY LIMIT EXCEEDED

IXC307I  STOP PATHOUT REQUEST FOR DEVICE 8030 COMPLETED
         SUCCESSFULLY:  RETRY LIMIT EXCEEDED

IXC409D  SIGNAL PATHS BETWEEN BP03 AND BP01 ARE LOST.  REPLY
         RETRY OR SYSNAME=SYSNAME OF THE SYSTEM TO BE REMOVED
```

Decision time!

- *CTCs are point to point connections*
- *If two systems can't communicate directly, one of them must be removed from the sysplex*

MPOR - 05 - 11

1.11 Losing the last or only CTC signalling path - 2

While you're looking for a coin to toss, the systems continue processing and XCF will attempt to restart the signalling path anyway (somebody might as well do something while you're dithering). Your options are:

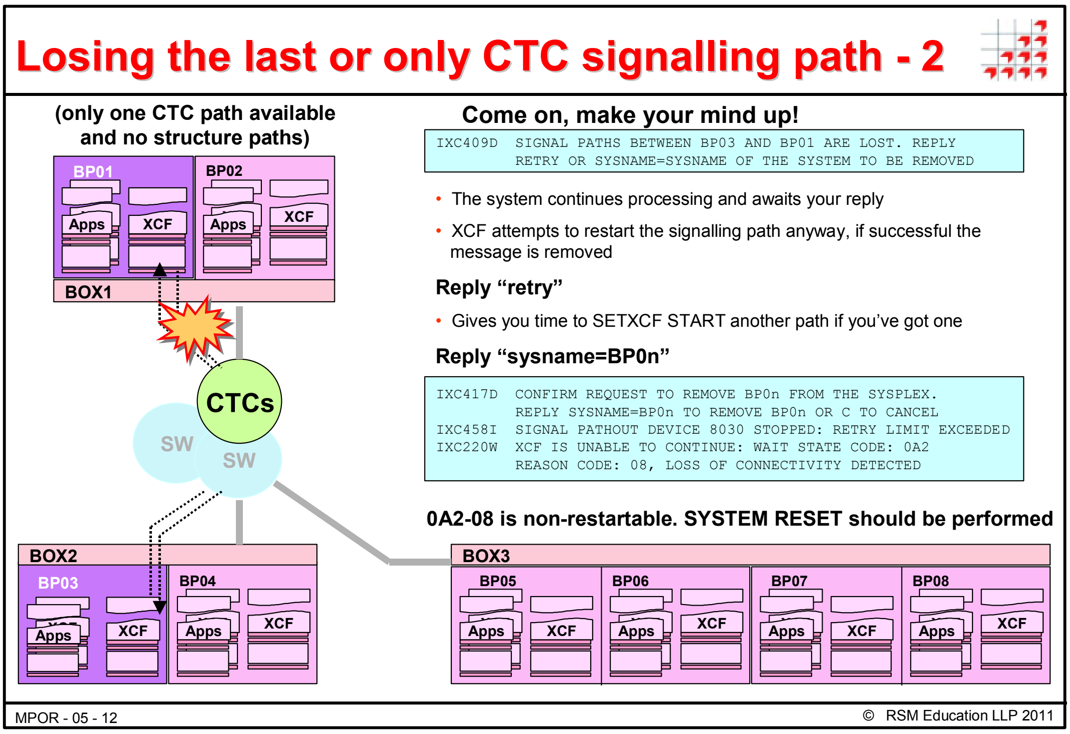
RETRY

This will allow you time to get somebody else to make the decision, or more practically, bring another path online if you have one. But then if you had another path available, you'd probably be using it anyway.

SYSNAME= BP0n

This will cause XCF to start removing BP0n from the sysplex. You will be asked to confirm your decision, you might see a confirmation of the original problem (the IXC458I in this example), and the end result is that system BP0n is placed into a **non-restartable wait state, code 0A2-08.**

You should SYSTEM RESET the dead system. You'll have to fix the path problem and re-IPL BP0n to bring it back into the sysplex.


Losing the last or only CTC signalling path - 2
(only one CTC path available
and no structure paths)
BP01
BP02
Apps
XCF
Apps
XCF
BOX1
CTCs
SW
SW
BOX2
BP03
BP04
Apps
XCF
Apps
XCF
BOX3
BP05
BP06
BP07
BP08
Apps
XCF
Apps
XCF
Apps
XCF
Apps
XCF
Come on, make your mind up!
IXC409D SIGNAL PATHS BETWEEN BP03 AND BP01 ARE LOST. REPLY
RETRY OR SYSNAME=SYSNAME OF THE SYSTEM TO BE REMOVED
• The system continues processing and awaits your reply
• XCF attempts to restart the signalling path anyway, if successful the message is removed
Reply "retry"
• Gives you time to SETXCF START another path if you've got one
Reply "sysname=BP0n"
IXC417D CONFIRM REQUEST TO REMOVE BP0n FROM THE SYSPLEX.
REPLY SYSNAME=BP0n TO REMOVE BP0n OR C TO CANCEL
IXC458I SIGNAL PATHOUT DEVICE 8030 STOPPED: RETRY LIMIT EXCEEDED
IXC220W XCF IS UNABLE TO CONTINUE: WAIT STATE CODE: 0A2
REASON CODE: 08, LOSS OF CONNECTIVITY DETECTED
0A2-08 is non-restartable. SYSTEM RESET should be performed
MPOR - 05 - 12
© RSM Education LLP 2011

1.12 Structure signalling path 'reconfiguration'

Now let's look at what happens if you're using **list structures** for signalling. In this first case, we've got a redundant configuration:

- two Coupling Facilities
- two signalling structures, one in each Coupling Facility
- two Coupling Facility channels from each system to each Coupling Facility.

Lose a CFC

If you lose a CFC connection to one of the structures, no problem. You'll get the IXL518I message as shown, and probably some kind of IOSnnnx message indicating the cause of the problem, but the system that was using this CFC will simply carry on using the structure via the other CFC.

Lose a Coupling Facility

If you lose one of the Coupling Facilities, either because you lose all access to it (both CFCs down) or because of actual Coupling Facility failure (both cases shown opposite), then still no problem in terms of signalling. The other Coupling Facility and structure are still available.

In this case though, you will certainly find applications start reporting failures if they had structures in the lost Coupling Facility.

XCF would also commence structure rebuild in this case, to recreate the lost signalling structure in the other Coupling Facility.

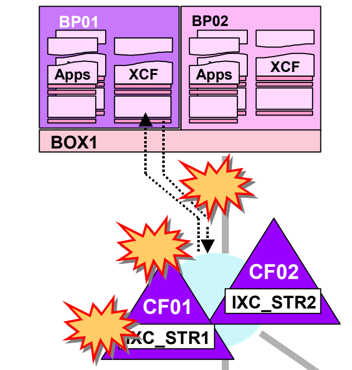
Losing a structure

If you have a **structure failure,** as indicated opposite, XCF would start the structure rebuild process. This time though, the structure would be rebuilt in the same Coupling Facility.

In all three cases, there is no outage and no systems are lost.

Structure signalling path 'reconfiguration'

(more than one list structure, Coupling Facility, and CFC to each CF)



1) CF Channel failure

```
IXL518I  PATH chpid IS NOW NOT OPERATIONAL TO CUID: CF cuid
          COUPLING FACILITY 009672.IBM.00.000020040104
          PARTITION: 1 CPCID: 00
```

(Probably accompanied by an IOSnnnx message)

2) Coupling Facility failure

```
IXL518I  (as above)
IXC518I  BP01 NOT USING COUPLING FACILITY (description) NAMED
          CF01 REASON: CONNECTIVITY LOST
```

or maybe

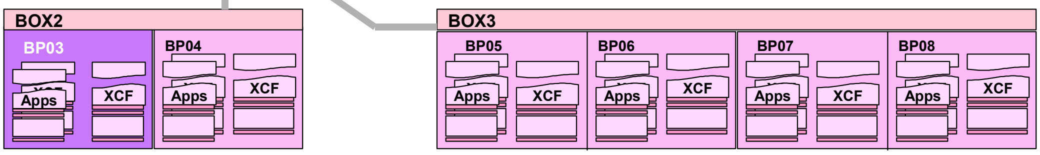
```
IXC519I  COUPLING FACILITY DAMAGE RECOGNIZED FOR COUPLING
          FACILITY (description) NAMED CF01
```

3) Structure failure

```
IXC467I  REBUILDING PATH STRUCTURE IXC_STR1. RSN: STRUCTURE
         FAILURE
```

(see "Sysplex Operations" topic, OPS00310, for remainder of messages)

In all cases, signalling continues using alternate facilities



MPOR - 05 - 13

1.13 Losing the only CFC to a signalling structure

So now let's look at those processes again, but this time in a configuration that doesn't have redundancy (so just one Coupling Facility, one CFC per system and one structure).

So if we lose the only CFC from BP01:

1) We start the same way as before, but now we don't have an alternate path as before, so we know that someone is going to be removed from the sysplex. Things get a little more complicated here though.

2) One question here is "who's actually affected?" It may be that the CFC is being shared via EMIF, in which case both BP01 and BP02 are hit. Let's assume the CFC is shared, and so both these systems have lost access to the structure.

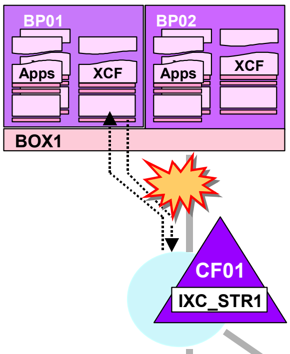
3) Unlike CTC paths, list structure 'paths' are used to connect multiple systems, and are not just point-to-point connections. So the IXC409D message will be issued now **for each pair of systems affected by the lost CFC.**

   What we're saying here is that each of our BP01 and BP02 systems has lost its signalling path to each other and all of BP03 through BP08. So on both BP01 and BP02, the IXC409D message is issued for each of the other seven systems, to ask which should be removed. Each message must be replied to, and of course you should reply BP01 or BP02 as appropriate in each case.

The sysplex will eventually be configured as systems BP03 through BP08.

Losing the only CFC to a signalling structure

(only one list structure, Coupling Facility, CFC to each CF and no CTC paths)



CF Channel failure

```
IXL518I  PATH chpid IS NOW NOT OPERATIONAL TO CUID: CF cuid
          COUPLING FACILITY 009672.IBM.00.000020040104
          PARTITION: 1 CPCID: 00
```

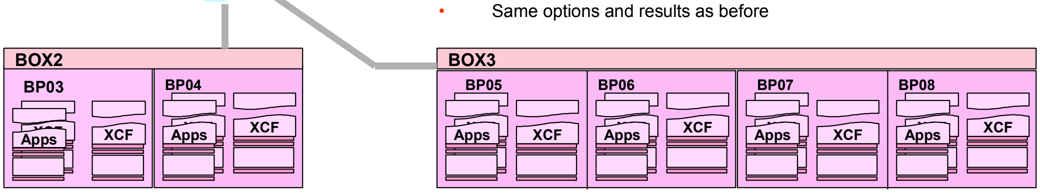
(Probably accompanied by an IOSnnnx message)

Who's affected?

- If the CFC definitions are shared via EMIF, all LPARs on the affected processor
- If dedicated, just the affected system

```
IXC519I  STOPPING PATH STRUCTURE IXC_STR1
         RSN: LOST CONNECTIVITY TO STRUCTURE

IXC409D  SIGNAL PATHS BETWEEN nnnn AND BP01 ARE LOST. REPLY
         RETRY OR SYSNAME=SYSNAME OF THE SYSTEM TO BE REMOVED
```

- IXC409D will be issued on BP01 (and BP02 if CFC shared) once for each system to which connectivity has been lost
- Same options and results as before



MPOR - 05 - 14

1.14 Losing the only CF (using a structure for signalling)

If we lose the only Coupling Facility, either due to the extremely unlikely event of losing all the CFCs from all the systems or due to a true Coupling Facility failure, then we have lost **all** signalling capability in the sysplex.

So we don't have a sysplex any more!

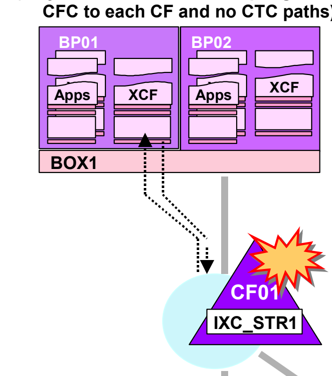
1) The IXC409D will be issued on **all systems**

2) **One system** can remain 'in the sysplex', i.e. up and running

3) All the others must be removed via the 0A2 wait state.

Losing the only CF (using a structure for signalling)

(only one list structure, Coupling Facility, CFC to each CF and no CTC paths)



Coupling Facility failure

```
IXL518I  (as before)
IXC510I  nnnn NOT USING COUPLING FACILITY (description) NAMED
         CF01 REASON: CONNECTIVITY LOST
```

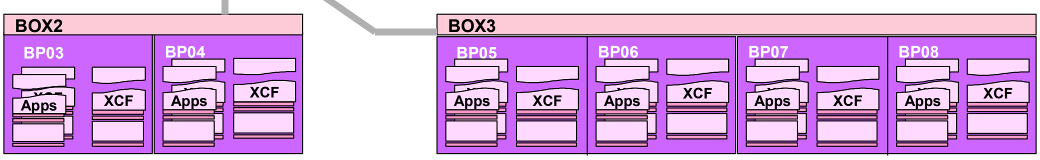
Or maybe

Who's affected?

- All systems!

```
IXC467I  STOPPING PATH STRUCTURE IXC_STR1
         RSN: LOST CONNECTIVITY TO FACILITY
IXC409D  SIGNAL PATHS BETWEEN nnnn AND nnnn ARE LOST. REPLY
         RETRY OR SYSNAME=SYSNAME OF THE SYSTEM TO BE REMOVED
```

- IXC409D will be issued on all systems
- Only one system allowed to remain active, all others must be removed via the 0A2 wait state



MPOR - 05 - 15

1.15 Losing the only signalling structure

If we lose the only **signalling structure,** but the actual Coupling Facility and all the CFCs are ok, XCF will simply initiate structure rebuild processing:

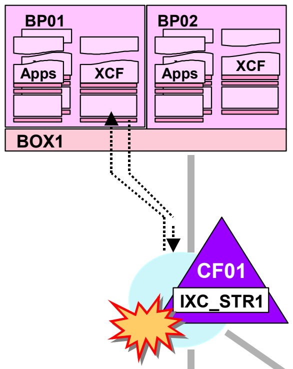
- all systems are affected

- but only temporarily, until the structure is rebuilt

If you see the second message, our old friend IXC467I again (the one with dozens of different message texts), then we've got a major problem and it will be as if we've lost the only Coupling Facility.

Losing the only signalling structure

(only one list structure, Coupling Facility, CFC to each CF and no CTC paths)



Structure failure

```
IXC467I  REBUILDING PATH STRUCTURE IXC_STR1. RSN: STRUCTURE
         FAILURE
```

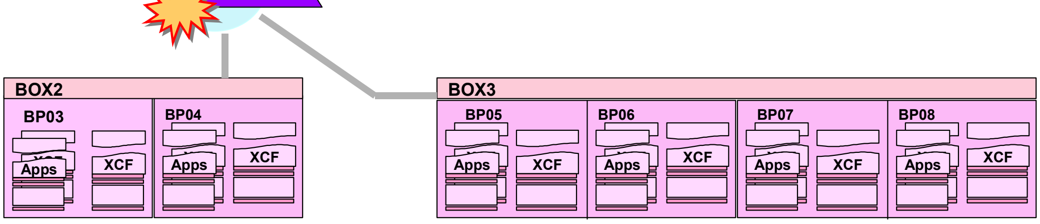
(see Sysplex Operation segment, OPS00310, for remainder of messages)

Who's affected?

- All systems, but only temporarily until the structure is rebuilt
- Of course, if you see this

```
IXC467I  STOPPING PATH STRUCTURE IXC_STR1
         RSN: REBUILD FAILED, UNABLE TO USE ORIGINAL
```

- Then it will be as if you've lost the Coupling Facility!



MPOR - 05 - 16

1.16 'Status update missing' conditions

And now for something completely different.

As we've said several times before, a z/OS system in a sysplex is seen as just one part of a co-operative environment. It must be ready to play its part at all times. We've just looked at signalling failures, because a loss of connectivity represents a **physical** inability to participate in a sysplex.

However, you may get a situation where a system is **logically** unable to participate because it has failed in some way. For example a z/OS system could drop into a disabled wait state or a tight disabled loop. If that were to happen and nothing were done, applications on other systems in the sysplex might also fail as they continued to attempt to communicate with their partner applications on the failed system, and so the problem might spread throughout the sysplex.

To prevent this happening, each system in the sysplex is expected to 'check in' within an installation specified interval. This is done by XCF updating the Couple Data Sets, and the time interval controlling this is controlled via the **INTERVAL** parameter in COUPLExx. It works like this:

- if a system doesn't update the Couple Data Sets within the COUPLExx **INTERVAL** value, the other XCFs raise a **status update missing** condition

- the **OPNOTIFY** parameter in the COUPLExx member indicates how long after the last status update the **IXC402D** message is issued, in the event of a status update missing condition being detected. Or, if you prefer, once the INTERVAL value has raised the alarm, XCF waits a further (OPNOTIFY minus INTERVAL) seconds before issuing the message.

What's happening is that the other XCFs are giving you the opportunity to correct any problems on the apparently failing system by replying with a further 'check in' interval value. And this value can be up to 86,400 seconds (one day)!

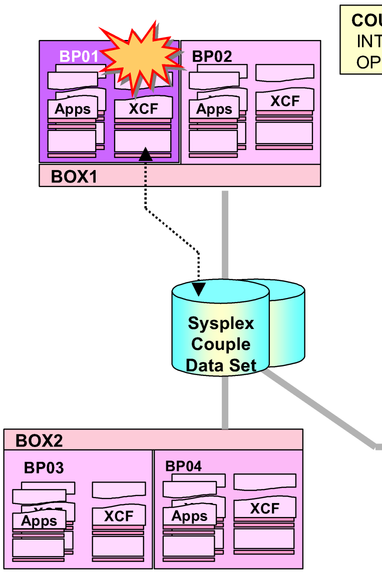
Reply "INTERVAL=sssss"

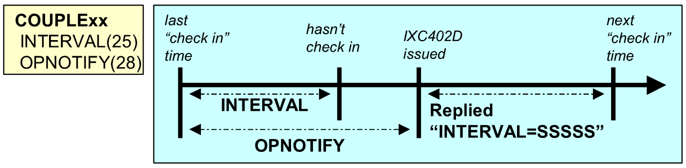
By specifying an interval value, you get the opportunity to recover the failing system.

It might be in a restartable wait state for example. It could be a loop that is resolved via a restart. It could be all manner of recoverable problems.

If you reply with an interval value, that system has 'sssss' seconds to resume status updating. If it hasn't done so at the end of that time, the IXC402D message is re-issued.

‘Status update missing’ conditions





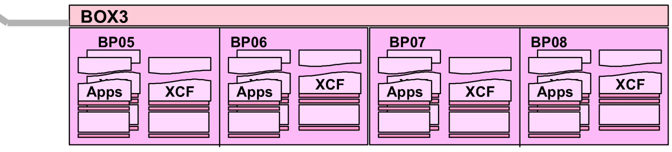
“Status update missing”

- If an XCF image fails to update the couple datasets within the INTERVAL time, the other XCFs raise a status update missing condition
- After the (OPNOTIFY-INTERVAL) time, the following message isssued:

```
IXC402D  BP01 LAST OPERATIVE AT hh:mm:ss. REPLY DOWN AFTER
         SYSTEM RESET OR INTERVAL=SSSSS TO SET A REPROMPT TIME
```

Implications?

- BP01 could theoretically be working fine, apart from XCF, but is probably in a disabled condition
- Could be restartable condition, or may need re-IPL!



MPOR - 05 - 17

Monitor detected ‘Stop’ status

```
IXC335I  17.04.41  DISPLAY XCF 479
 SYSPLEX RSMPLX
 SYSTEM      TYPE   SERIAL   LPAR     STATUS        TIME          SYSTEM STATUS
 RSMA        2086   722D      03    06/13/2010    17:04:40      ACTIVE TM=SIMETR
 RSMB        2086   722D      04    06/13/2010    17:04:15    MONITOR-DETECTED STOP
```

MPOR - 05 - 18

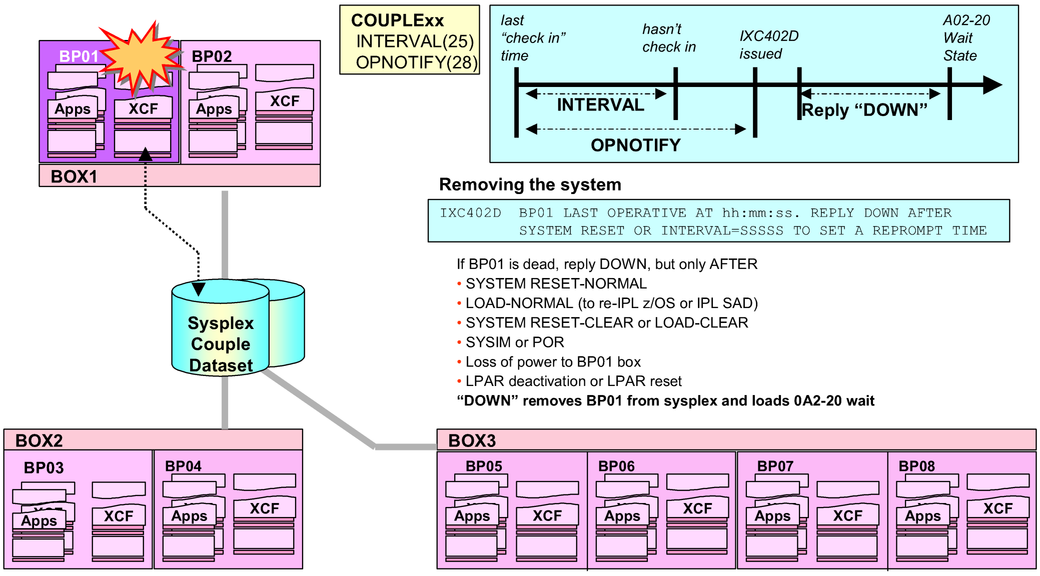
1.17 Removing the system and replying "down"

Of course, it may be that the system concerned is completely unrecoverable.

In this case, you should reply **DOWN** to the ICX402D message, ***but only after performing a system reset, or any of the other functions listed opposite, on the failing system.***

Replying **DOWN** causes the other XCFs to remove the failing system from the sysplex. This means placing it into an **0A2-20 non restartable wait state** (assuming it has been system reset, rather than, for example IPLed for a stand alone dump). It also means marking it as no longer part of the sysplex in the Couple Data Set status data. This stops all the XCFs checking on its status, and ensures that no other XCF attempts to communicate with this system in future.

Removing the system and replying "down"



MPOR - 05 - 19

D XCF after system has been removed

```
IXC335I   17.17.02   DISPLAY XCF 559
SYSPLEX RSMPLX
SYSTEM    TYPE   SERIAL  LPAR   STATUS          TIME         SYSTEM STATUS
RSMA      2086   722D     03   06/13/2010 17:17:01            ACTIVE TM=SIMETR
```

MPOR - 05 - 20

1.18 SPINTIME and INTERVAL

The **INTERVAL** value in COUPLExx represents the time period after which, if a system has not updated the Couple Data Sets, it is considered to be inoperative. It is an 'external' value, in that it refers to a z/OS system having problems that are considered to be stopping it from working with the outside world.

There is a similar time value used to deal with a particularly disabling problem internally, the **SPINTIME** parameter in the **EXSPATxx** member of SYS1.PARMLIB. This determines how long a CPU inside a z/OS system is allowed to remain in a disabled spin loop. If a CPU spins for longer than the SPINTIME value, an **excessive spin loop** condition is raised.

The EXSPATxx member determines what to do in these conditions. The default is to allow another SPINTIME interval first, and then if the interval is exceeded again, **abend** the unit of work running on the spinning processor.

The relationship between SPINTIME and INTERVAL

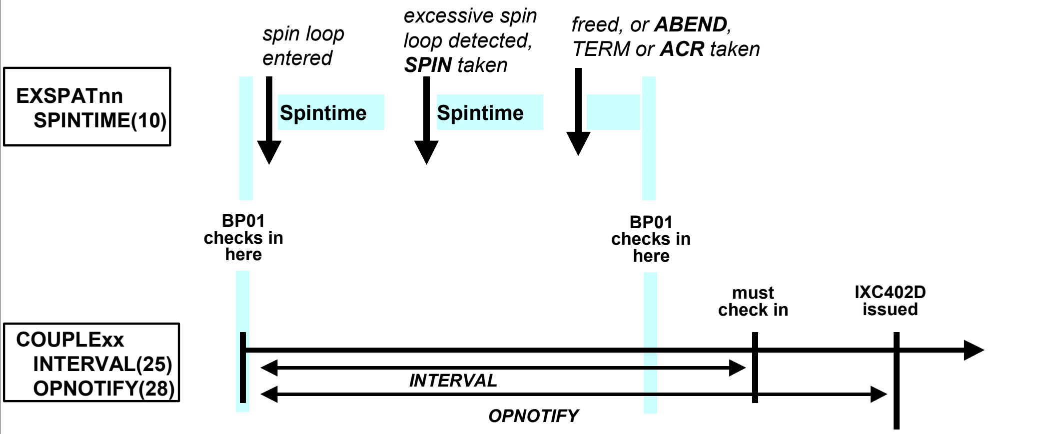
If a z/OS system has one or more processors stuck in a disabled spin loop, it may not get a chance to get out and update the Couple Data Sets. So the SPINTIME interval, which detects this problem, should be smaller than the INTERVAL value. The defaults are:

- **SPINTIME(10)** and **INTERVAL(25)** if the z/OS image is running in native mode or in a dedicated LPAR

- **SPINTIME(40)** and **INTERVAL(85)** if the z/OS image is running in a shared LPAR

If you change the SPINTIME value, **or the SPINRCY options** (i.e. to allow more than two SPINTIME intervals per spin), make sure you adjust INTERVAL accordingly.

SPINTIME & INTERVAL



SPINTIME & INTERVAL

- ***SPINTIME is an 'internal' value, it represents problems 'inside' the system***
- ***INTERVAL is an 'external' value, it represents a problem at the sysplex level***
- ***SPINTIME should be less than INTERVAL***

MPOR - 05 - 21

1.19 System Isolation techniques

How a system is isolated (removed from the sysplex) differs depending on the situation.

Connectivity failures

If BP01, for example, loses its last or only connection to one or more systems in the sysplex it will be isolated and placed into the 0A2 wait state.

In this case, it is actually a functioning z/OS system, it just can't communicate with the outside world. But it is still operational, and the XCF on this system will itself know that it has lost connectivity, so the XCFAS on this system loads the wait state.

Status update missing

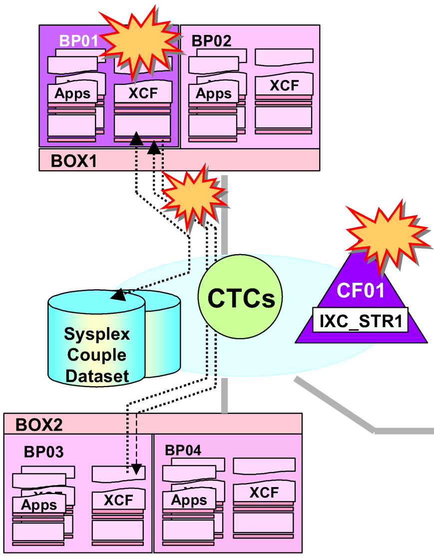
If poor old BP01 goes into a disable condition and misses its update interval, it will also be isolated via the 0A2 wait state. But now this is not a functioning system, so it can't do this for itself.

So in this second case, system isolation is done from the outside, **via the Coupling Facility:**

- another system's XCF asks the Coupling Facility to isolate BP01
- the Coupling Facility sends an isolate request to the channel subsystem on the target system
- the channel subsystem does an I/O reset (like a system reset) and then loads the wait state on all the processors

Because this is done by the channel subsystem, it doesn't need a functional z/OS environment.

System Isolation techniques



Connectivity failures

- If BP01 loses its last or only connection to one or more systems in the sysplex it will be isolated
- BP01 is a working system, it's just lost communications with the sysplex
- So XCF on BP01 will load 0A2 wait state for itself

Status update missing

- If BP01 goes into a disabled condition, it will miss its update interval and will be isolated
- But in this case ***BP01 is dead and can't post its own wait state***
- System isolation:
  - is performed via the Coupling Facility from another system in the sysplex
  - is done via the channel subsystem on the target system (BP01)

MPOR - 05 - 22

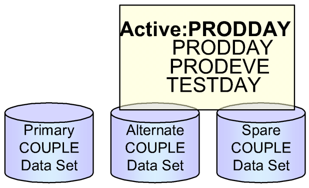
1.20 SFM and ARM

The Sysplex Failure Manager and Automatic Restart Manager are both additional XCF features that can be set up to assist in automating error handling in the sysplex. Both are optional, in that the sysplex will run quite happily without them, and both are controlled via policies in their respective Couple Data Sets.

Different SFM and ARM requirements might exist at different times in the sysplex, and so policies can be changed dynamically in the usual way.

SFM & ARM

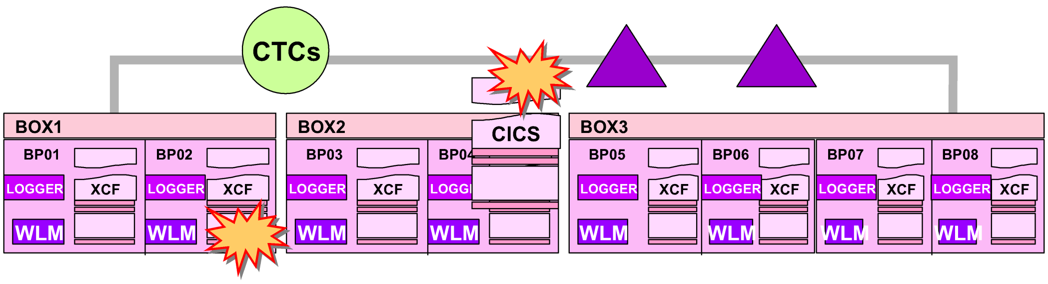
Sysplex Failure Manager deals with XCF level sysplex failures
Automatic Restart Manager restarts failed jobs



SFM and ARM
Couple Data Sets and policies

Related components

- **Implemented via XCF**
- Policies for dealing with failures in the sysplex
- Different policies for different workloads(overnight, etc.)
- Policies can be switched with SETXCF START,POLICY



MPOR - 05 - 23

1.21 The Sysplex Failure Manager (SFM)

Dealing with connectivity failures and status update missing conditions so far has been a purely manual process, but it can be automated via the **Sysplex Failure Manager.**

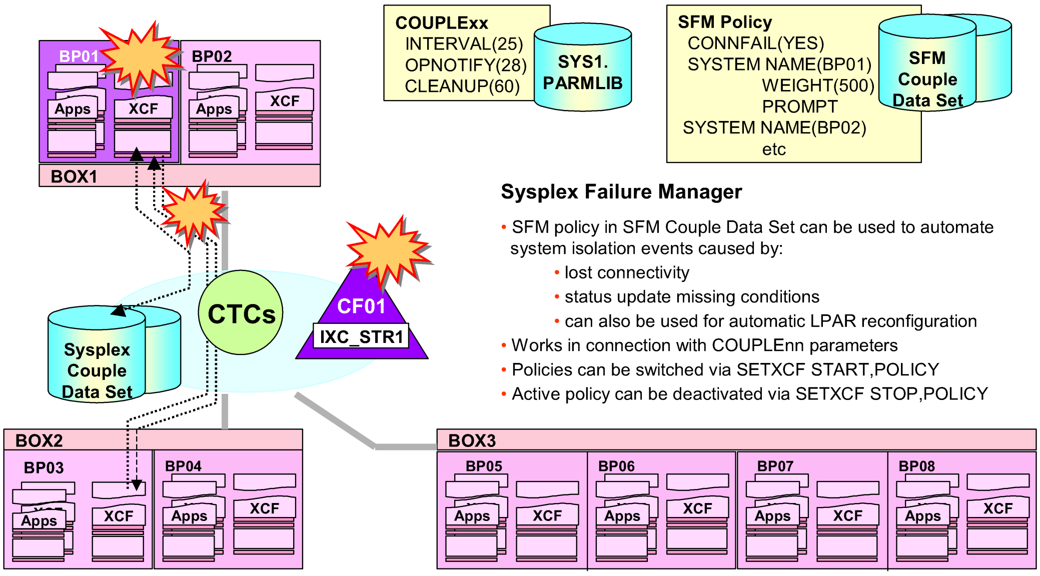
SFM is controlled via an SFM policy in an SFM Couple Data Set, and works in conjunction with the parameters in COUPLExx.

SFM policies and Couple Data Sets can of course be switched via the SETXCF command.

LPAR reconfiguration

SFM can also be used to control automatic LPAR reconfiguration activities in the event of system isolation events. This is not discussed on this class as few sites use these techniques.

The Sysplex Failure Manager (SFM)



Sysplex Failure Manager

- SFM policy in SFM Couple Data Set can be used to automate system isolation events caused by:
  - lost connectivity
  - status update missing conditions
  - can also be used for automatic LPAR reconfiguration
- Works in connection with COUPLEnn parameters
- Policies can be switched via SETXCF START,POLICY
- Active policy can be deactivated via SETXCF STOP,POLICY

MPOR - 05 - 24

1.22 SFM policy options

The SFM policies are created by the IXCMIAPU utility as usual. The basic idea is that you can define parameters for each system to control how that system should be handled in the event of connectivity and status update missing failures. The options available are as follows.

1) **CONNFAIL** is very important. It says whether or not SFM should automatically handle connectivity failure processing, by using the system **weights** (below).

2) For each system, by name, you can then define the SFM options. In the first definition shown, we're specifying defaults to be used for all systems **"NAME(*)".**

   - We then code what is now a default **WEIGHT** value. The weights are used to determine the **relative importance** of each system when deciding who to remove during *connectivity failure* processing.

   - We're also coding the **PROMPT** parameter as a default. This is used for *status update missing* conditions and causes the IXC402D message to be issued for all systems in those circumstances.

3) Having specified our defaults, we now create our exceptions. In this case, BP01 is given a WEIGHT of 500, which means this system is 500 times more important than any system its connected to in the event of a connectivity failure involving this system

4) For BP02, we want to do something other than just prompt the operators in the event of a status update missing condition. Instead of issuing the IXC402D message, we will:

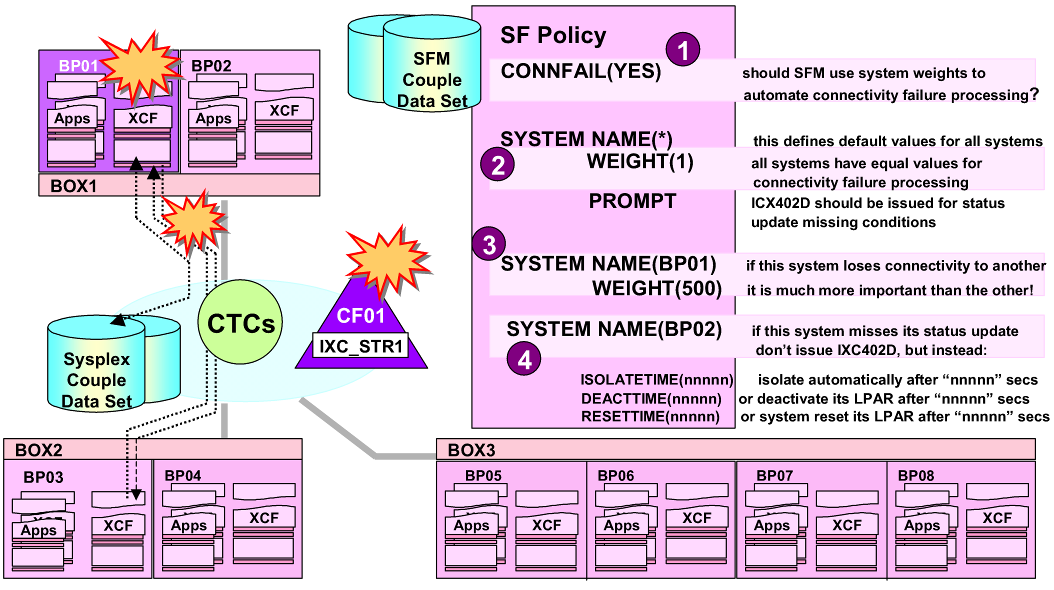
   - **isolate** the system automatically after 'nnnnn' seconds,

   - or **deactivate its LPAR** automatically after 'nnnnn' seconds,

   - or **system reset its LPAR** automatically after 'nnnnn' seconds.

So, let's look at how some of these options work.

SFM policy options



MPOR - 05 - 25 © RSM Education LLP 2011

Starting & stopping the SFM policy

```
SETXCF START,POLICY,POLNAME=SFMPOL1,TYPE=SFM

IXC616I SFM POLICY SFMPOL1 INDICATES CONNFAIL(YES) FOR SYSPLEX RSMPLX
IXC602I SFM POLICY SFMPOL1 INDICATES ISOLATETIME(0) 485
SSUMLIMIT(25) FOR SYSTEM RSMA FROM THE DEFAULT POLICY ENTRY.
IXC609I SFM POLICY SFMPOL1 INDICATES FOR SYSTEM RSMA A SYSTEM WEIGHT OF
5 SPECIFIED BY POLICY DEFAULT
IXC614I SFM POLICY SFMPOL1 INDICATES MEMSTALLTIME(NO) FOR SYSTEM RSMA AS
 SPECIFIED BY SYSTEM DEFAULT
IXC601I SFM POLICY SFMPOL1 HAS BEEN STARTED BY SYSTEM RSMA

 TYPE: SFM
      POLNAME:        SFMPOL1
      STARTED:        06/13/2010 17:06:55
      LAST UPDATED:   06/13/2010 10:36:34


 SETXCF STOP,POLICY,TYPE=SFM
 IXC607I SFM POLICY HAS BEEN STOPPED BY SYSTEM RSMA

 TYPE: SFM
 POLICY NOT STARTED
```

MPOR - 05 - 26 © RSM Education LLP 2011

1.23 SFM processing for connectivity failures

In the example opposite:

1) We have a CTC only signalling configuration, with only one pair of connections between each system.

2) The path from BP03 to BP01 fails

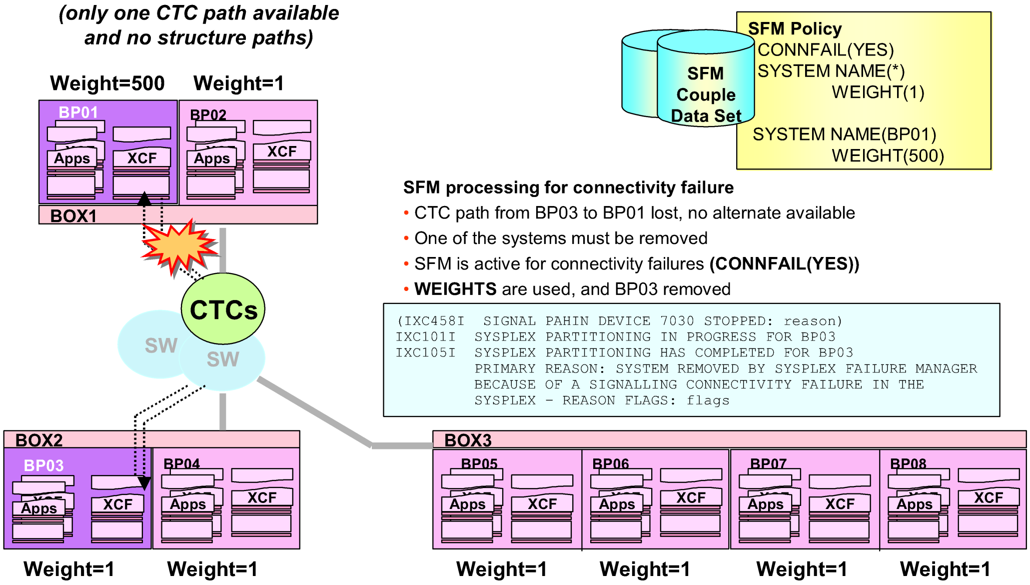
3) No alternate path is available, so one of the two systems must be removed.

4) We have an active SFM policy which includes the **CONNFAIL(YES)** setting, so SFM takes over.

5) SFM checks the weights of BP01 and BP03, so it looks like BP03 is the loser here.

6) SFM removes BP03, and issues the messages to indicate what has happened

SFM processing for connectivity failures



MPOR - 05 - 27

Displaying SFM parameters

```
SFM Active

 INTERVAL    OPNOTIFY      MAXMSG    CLEANUP      RETRY   CLASSLEN
   85           88          2000       15          10       956

 SSUM ACTION  SSUM INTERVAL  SSUM LIMIT     WEIGHT   MEMSTALLTIME
   ISOLATE        0               25          5          NO
```

MPOR - 05 - 28

1.24 CF signalling, connectivity failures and SFM's weights

Be careful when coding SFM weight values if you are signalling via Coupling Facility list structures.

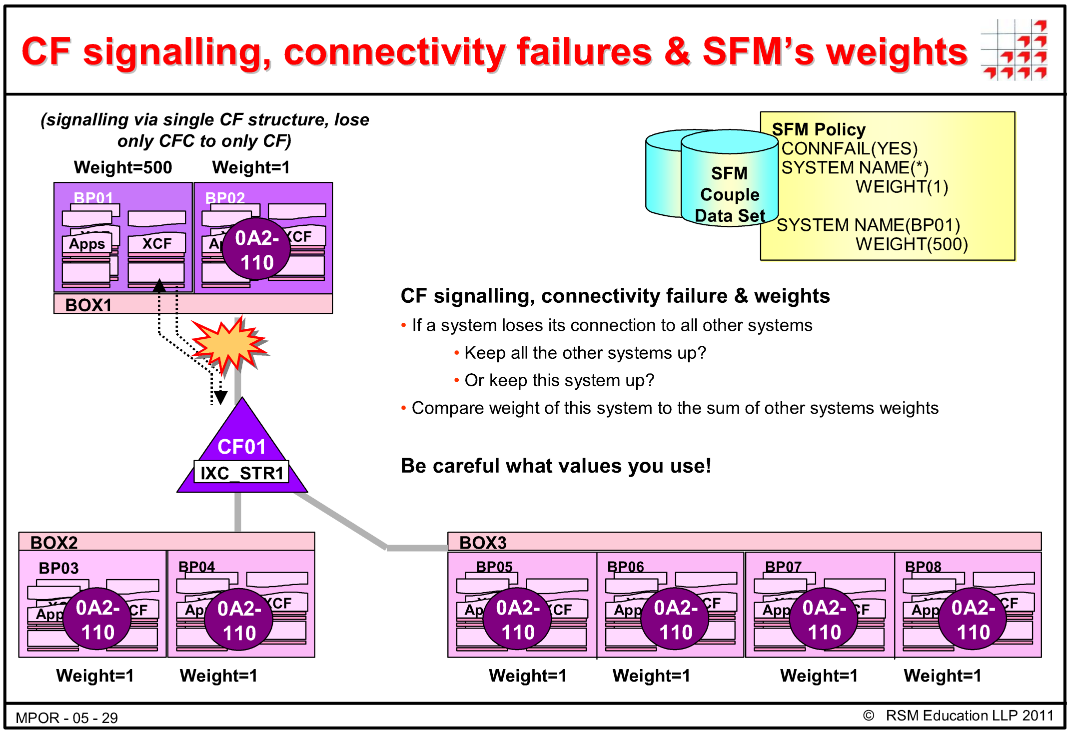
In the example opposite, BP01 has a weight of 500 and all the other systems have a weight of 1, as before. Now, however, the systems are signalling via a list structure in the only Coupling Facility. BP01 loses its only CFC link to that Coupling Facility and therefore loses its connection to all the other systems. The decision now is "should we keep BP01 and kill all the other systems or keep all the others and kill BP01?". If you have an active SFM policy with CONNFAIL(YES) coded, the weight calculation is to compare the weight of BP01 against the sum of the weights of all the other systems. In this example, BP01 remains active and all the others are isolated out of the sysplex.

Which is a bit unfortunate as the only system remaining in the sysplex is also the only one that can't access the Coupling Facility!

Mixed signalling environments

If CTCs were being used for signalling, BP01 having a weight of either 500 or 5 would have the same effect, if all other systems had a weight of 1. But in a CF signalling environment, these two different weights have rather different effects.

When setting the weights in a sysplex that uses both CTCs and CF structures for signalling, make sure you set the weights appropriately, that is to say remembering how they work for the CF signalling failures.


CF signalling, connectivity failures & SFM's weights
(signalling via single CF structure, lose only CFC to only CF)
Weight=500
Weight=1
BP01
BP02
Apps
XCF
0A2-110
BOX1
CF01
IXC_STR1
SFM Couple Data Set
SFM Policy
CONNFAIL(YES)
SYSTEM NAME(*)
WEIGHT(1)
SYSTEM NAME(BP01)
WEIGHT(500)
CF signalling, connectivity failure & weights
• If a system loses its connection to all other systems
• Keep all the other systems up?
• Or keep this system up?
• Compare weight of this system to the sum of other systems weights
Be careful what values you use!
BOX2
BP03
BP04
BOX3
BP05
BP06
BP07
BP08
Weight=1
Weight=1
Weight=1
Weight=1
Weight=1
Weight=1
MPOR - 05 - 29
© RSM Education LLP 2011

1.25 SFM processing for status update missing

In this example, BP01 enters a status update missing condition

1) By default, our active SFM policy issues the IXC402D message. This is done after the COUPLExx OPNOTIFY period expires.

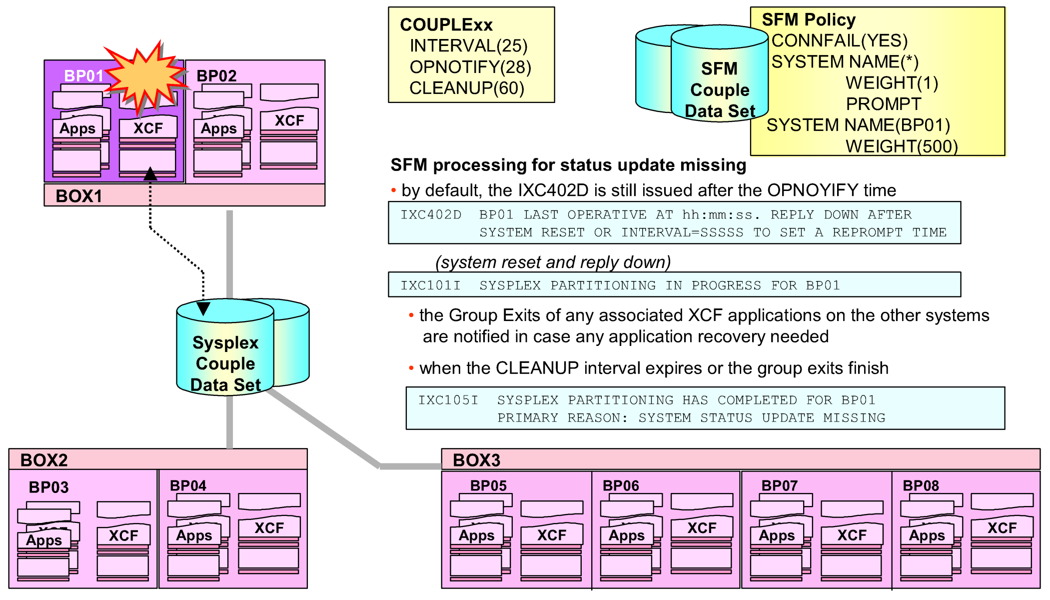
2) BP01 can't be restarted, so SYSTEM RESET is performed and the operators reply DOWN.

3) XCF starts the partitioning process and issues IXC101I.

4) XCF then **notifies the group exits** of all the members on BP02 through BP08 of those XCF groups that also had members on BP01. The idea here is that applications might need to 'clean up' before BP01 is removed from the sysplex.

5) When all the group exits have responded, or the COUPLExx **CLEANUP** interval expires, **whichever comes first,** BP01 will be placed into the 0A2 wait state and IXC105I issued.

SFM processing for status update missing



MPOR - 05 - 30

Sysplex partitioning

SFM in action

```
IXC101I SYSPLEX PARTITIONING IN PROGRESS FOR RSMB REQUESTED BY
XCFAS. REASON: SFM STARTED DUE TO STATUS UPDATE MISSING
*22 IXC102A XCF IS WAITING FOR SYSTEM RSMB DEACTIVATION. REPLY DOWN WHEN
 MVS ON RSMB HAS BEEN SYSTEM RESET

RESPONSE=RSMA
 IXC335I  17.14.23  DISPLAY XCF 498
 SYSPLEX RSMPLX
 SYSTEM   TYPE    SERIAL   LPAR      STATUS      TIME          SYSTEM STATUS
 RSMA     2086     722D     03     06/13/2010  17:14:22     ACTIVE TM=SIMETR
 RSMB     2086     722D     04     06/13/2010  17:11:57     BEING REMOVED - RSMA
```

MPOR - 05 - 31

1.26 SFM processing for status update missing - 2

In this example:

1) BP01 enters a status update missing condition

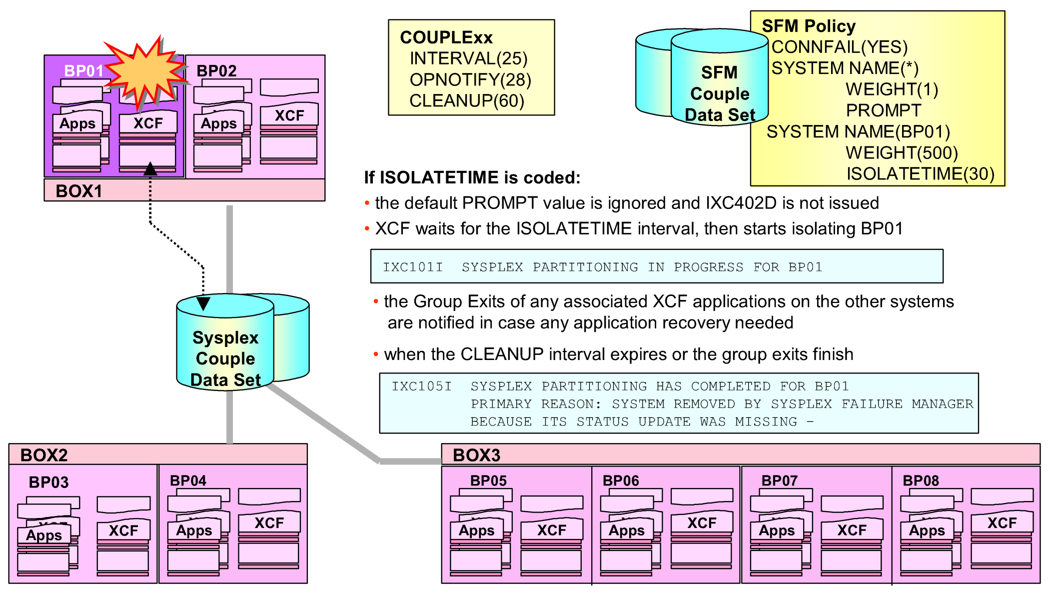
2) The active SFM policy has a definition for BP01 that precludes the issuing of the IXC402D prompt, so the OPNOTIFY value is ignored and the message is not issued.

3) Instead, **ISOLATETIME(30)** is specified, so XCF waits 30 seconds, issues IXC101I and starts to isolate BP01

4) XCF then notifies the group exits of all the associated members as before.

5) And again, when all the group exits have responded, or the COUPLExx **CLEANUP** interval expires, BP01 will be placed into the 0A2 wait state and IXC105I issued.

SFM processing for status update missing - 2



MPOR - 05 - 32

1.27 SFM, system isolation

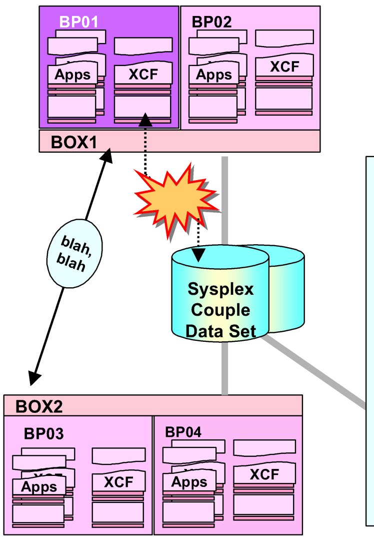
XCF now includes an extra check when a status update missing condition is detected.

- If SFM is active, and
- a system is defined to SFM with the ISOLATETIME parameter, and
- that systems has missed its status update interval, **but it is still detected as communicating with other systems,**

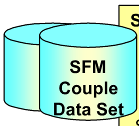
then messages IXC427A and possibly IXC426D will be issued rather than XCF commencing the isolation process according to the ISOLATETIME value.

XCF is ‘giving the system another chance’ by allowing it another INTERVAL(nn) seconds to check in. If you reply to the IXC426D message (if issued), you can force isolation for the offending system if you know it is actually dead and will need to be re-IPLed anyway.

SFM – system isolation



COUPLExx
INTERVAL(25)
OPNOTIFY(28)
CLEANUP(60)



SFM Policy
CONNFAIL(YES)
SYSTEM NAME(*)
WEIGHT(1)
PROMPT
SYSTEM NAME(BP01)
WEIGHT(500)
ISOLATETIME(30)

ISOLATETIME

- If the system has missed a status update and is not signalling, things proceed as just described
- If the system has missed a status update but is still signalling other systems, it is not isolated immediately

```
IXC427A  SYSTEM BP01 HAS NOT UPDATED STATUS SINCE hh:mm:ss BUT IS SENDING
         XCF SIGNALS. XCF SYSPLEX FAILURE MANAGEMENT WILL REMOVE BP01 IF
         NO SIGNALS ARE RECEIVED WITHIN A SECOND INTERVAL
IXC426D  SYSTEM BP01 IS SENDING SIGNALS BUT NOT UPDATING STATUS.
         REPLY SYSNAME BP01 TO REMOVE THE SYSTEM [OR R TO RETRY]
```

- Processing continues meanwhile
- If signalling stops, BP01 will be isolated as before
  - reply BP01 will isolate BP01
  - do nothing (or reply R), BP01 remains in sysplex for a further interval

MPOR - 05 - 33

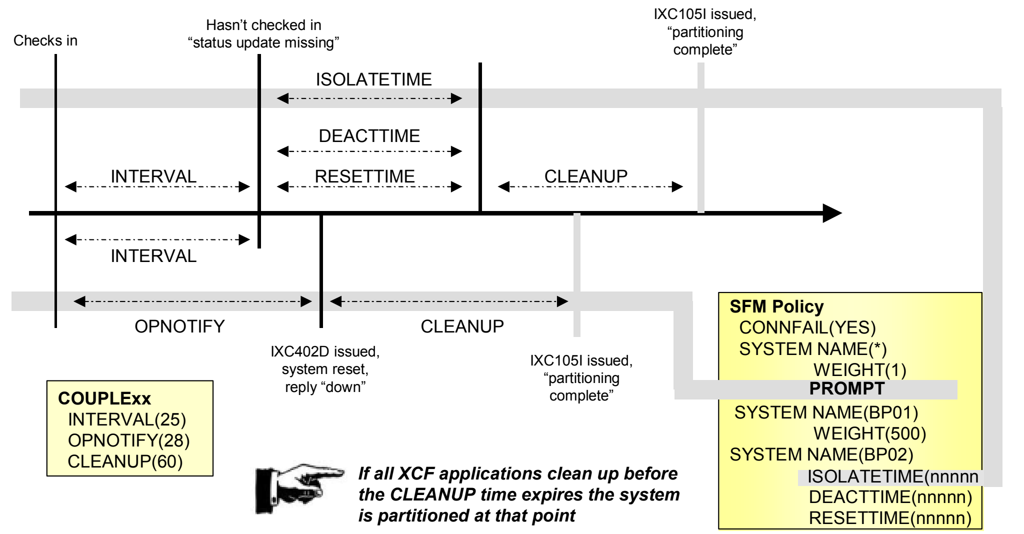
1.28 Time interval relationships with SFM

A summary of the relationships between the various COUPLExx and SFM policy options is shown opposite.

The COUPLExx values are the defaults for a native or dedicated LPAR environment.

The defaults for the SFM policy options are as shown on the SYSTEM NAME(*) statement.

Time interval relationships with SFM



If all XCF applications clean up before the CLEANUP time expires the system is partitioned at that point

MPOR - 05 - 34

1.29 The SFM environment

1) You can display details of the SFM environment as shown opposite.

2) You can also change those values supplied by the COUPLExx member via the SETXCF COUPLE command. To change the values that come from the SFM policy you must change the SFM policy itself.

3) Stopping an SFM policy doesn't stop system isolation occurring in the event of connectivity failures and status update missing conditions. It just stops SFM from automating it.

The SFM environment

COUPLExx
INTERVAL(25)
OPNOTIFY(28)
CLEANUP(60)

SFM Couple Data Set

SFM Policy
CONNFAIL(YES)
SYSTEM NAME(*)
WEIGHT(1)
PROMPT
SYSTEM NAME(BP01)
WEIGHT(500)
ISOLATETIME(30)

DISPLAY XCF,POLICY,TYPE=SFM

```
IXC364I  20:22:04 DISPLAY XCF
         TYPE: SFM
              POLNAME:              SFMPOL99
              STARTED:              05/25/97 18:03:22
              LAST UPDATED:         05/25/97 18:03:22
```

DISPLAY XCF,COUPLE

```
IXC357I 20:28:14 DISPLAY XCF
        SYSTEM BP01 DATA
         INTERVAL    OPNOTIFY   MAXMSG   CLEANUP     RETRY   CLASSLEN
            25          28        500        60        10       956

         SSUM ACTION   SSUM INTERVAL   WEIGHT
           ISOLATE            25         500
```

Can be changed via SETXCF

Can only be changed by changing SFM policy

SETXCF COUPLE,INTERVAL=nn [,OPNOTIFY=nn, CLEANUP=nn]

```
IXC309I  SETXCF COUPLE,INTERVAL REQUEST WAS ACCEPTED
```

SETXCF STOP,POLICY,TYPE=SFM

```
IXC607I  SFM POLICY HAS BEEN STOPPED BY SYSTEM BP01
```

SETXCF START,POLICY,TYPE=SFM,POLNAME=SFMPOL77

```
IXC601I  SFM POLICY HAS BEEN STARTED BY SYSTEM BP01
```

MPOR - 05 - 35

1.30 Enabling SFM, switching SFM data sets

Some information about the SFM environment.

Activating SFM

As usual, you don't have to re-IPL the sysplex to activate SFM, as it can be done via SETXCF, as shown opposite.

1) Use SETXCF COUPLE to bring the SFM Couple Data Sets into use and SETXCF START, POLICY to activate the SFM policy. These commands have sysplex scope so can be issued on any system.

2) All systems in a sysplex must have access to these Couple Data Sets for SFM to become active.

3) The COUPLExx member should then be updated to reflect the new environment.

SFM status retained across IPLs

If you switch SFM policies, this is logged in the SFM Couple Data Sets just like we saw in earlier topics with CFRM. So if you shut down a system, or even the whole sysplex, when systems re-IPL and rejoin the sysplex they will connect to the correct SFM Couple Data Sets via the COUPLExx member and the last used policy will still be active.

Switching SFM Couple Data Sets

You can switch SFM Couple Data Sets via the SETXCF COUPLE,PSWITCH and ACOUPLE commands in the normal way.

IPLing after an SFM CDS switch

If a system IPLs into an active sysplex and has the wrong SFM Couple Data Sets specified in its COUPLExx member, the other systems' SFM just tell the incoming system which data sets are actually in use right now, just like we saw with the sysplex Couple Data Sets back in the **Sysplex Operations** chapter earlier.

Enabling SFM, switching SFM data sets

Turn SFM on dynamically

SETXCF COUPLE,TYPE=SFM,PCOUPLE=
SETXCF COUPLE,TYPE=SFM,ACOUPLE=
SETXCF START,POLICY,TYPE=SFM,POLNAME=

- *the above commands have sysplex scope if all systems have access to data sets*
- *all systems must have access for SFM to become active*
- *COUPLExx should be updated to reflect SFM data sets*

SFM status retained across IPLs

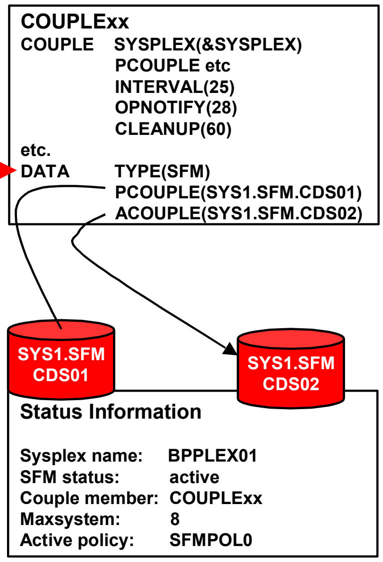
- *if you shut down a system or the sysplex, last used data sets and policy activated on re-IPL*

Switching SFM data sets

SETXCF COUPLE,PSWITCH,TYPE=SFM
SETXCF COUPLE,ACOUPLE(SYS1.SFM.CDS03),TYPE=SFM

Re-IPL one system (e.g. BP04) after SFM CDS switch

- *no problems rejoining sysplex, even though COUPLEnn specifies ‘wrong’ SFM Couple Data sets*
- *SFM on BP04 is told by other SFMs that CDS02 and CDS03 currently in use instead*



MPOR - 05 - 36

1.31 Other SFM considerations

One thing that is different with SFM to the situation with the sysplex Couple Data Sets in the **Sysplex Operations** chapter is when you bring shut the entire sysplex down after switching SFM Couple Data Sets.

Re-IPL sysplex after SFM CDS switch

If a system IPLing to initialize a sysplex specifies a different set of SFM Couple Data Sets to those currently marked as active, then the sequence of messages shown opposite is issued to ultimately ask *"do you want to use the ones from the COUPLExx definition or the last ones used?"*

Also, if SFM was previously active when the sysplex was shut down, it will be active immediately on each system as it rejoins the sysplex, so there will be additional IPL messages from SFM.

VARY XCF,sysname,OFFLINE with SFM

If SFM is active in the sysplex and a system is defined to SFM with an ISOLATETIME(nn) parameter, then if you issue a VARY XCF,sysname, OFFLINE for that system SFM takes over and isolates that system automatically, **and you won't be asked to confirm the vary command.** Be afraid, be very afraid!

Other SFM considerations

Re-IPL sysplex after SFM data sets switched

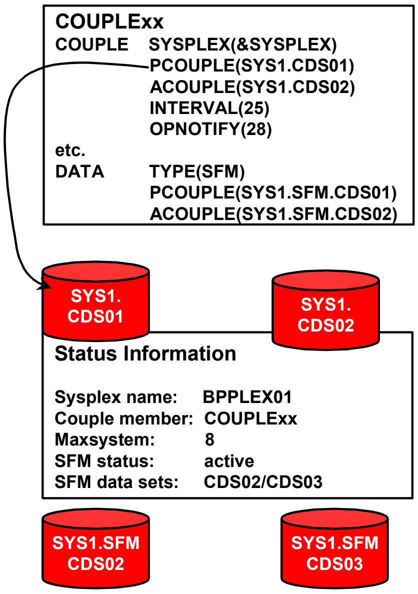
- *SFM switched to CDS02/03 from CDS01/02*
- *Shutdown whole sysplex, re-IPL first system*

```
IXC2871   THE COUPLE DATASETS SPECIFIED IN COUPLEnn ARE
          INCONSISTENT WITH THOSE LAST USED FOR SFM
IXC2881   COUPLE DATASETS SPECIFIED IN COUPLEnn FOR SFM ARE
          PRIMARY:   SYS1.SFM.CDS01 ON VOLSER volser
          ALTERNATE: SYS1.SFM.CDS02 ON VOLSER volser
IXC2881   COUPLE DATASETS LAST USED FOR SFM ARE
          PRIMARY:   SYS1.SFM.CDS02 ON VOLSER volser
          ALTERNATE: SYS1.SFM.CDS03 ON VOLSER volser
IXC289D   REPLY U TO USE THE DATA SETS LAST USED FOR SFM
          OR C TO USE THE DATA SETS SPECIFIED IN COUPLEnn
```

- *Also, there are a bunch of SFM confirmation messages issued on each system at IPL*

Varying a system offline with SFM active

- *If SFM is active...*
- *...and ISOLATION is specified for the target system*
- *V XCF,sysname,OFFLINE will result in automatic isolation for that system, i.e. no IXC 102A (reply "down") message is issued*



MPOR - 05 - 37

1.32 Clocks, clocks and more clocks

The 9037 External Timer Reference (ETR) facility, or sysplex timer, is a required element in a parallel sysplex. It ensures that all the TOD clocks on the participating machines in the sysplex are kept in step. So let's look at the clock environment next. We start at the point where you first install and set up the sysplex timer.

1) Once the sysplex timer network is installed and connected to the appropriate machines, and the sysplex timer consoles are attached and initialised (these topics are not covered on this course), the initial time, date and timezone offset are set either manually from the 9037 console or via an External Time Source via the network.

2) The 9672 machines have a Support Element Battery Operated Clock (BOC), and this is initially set manually from the HMC when the 9672 is powered up.

3) If the 9672 detects a working 9037 attachment on power up, then the BOC is updated from the 9037 time signal, thus ensuring these two clocks are synchronised.

4) In the 9672, the CPC contains the Time Of Day (TOD) clock. On power up, the TOD is set from the SE's BOC, so at this point the physical TOD (PTOD) is also synchronised with the 9037.

5) In LPAR mode, PR/SM maintains a Logical TOD (LTOD) for each partition. This is set from the PTOD as each LPAR is activated. As each LPAR is IPLed, the IPLing z/OS system checks the CLOCKnn member and checks the **ETRMODE** value.

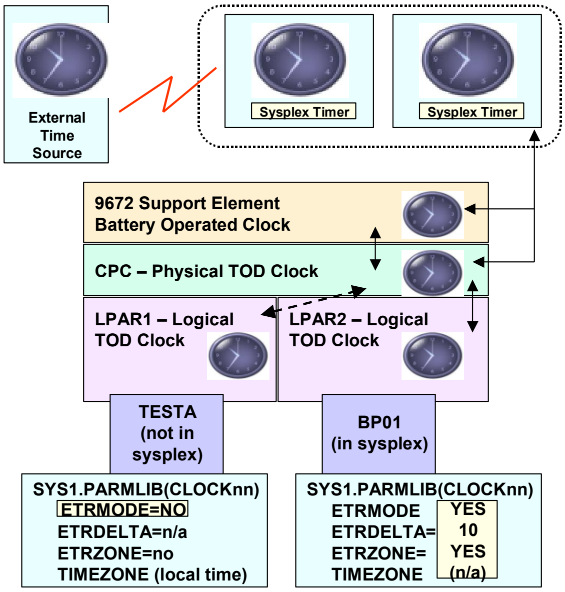
   ETRMODE=NO If a z/OS system is IPLed with ETRMODE=NO, then that z/OS system either sets its time according to the LTOD or issues the operator **SET TIME/DATE** prompt, depending on the PROMPT parameter in CLOCKnn. This system's LTOD will not subsequently remain synchronised with the 9037, and depending on the CLOCKnn **PROMPT** value, the LTOD value may be fixed for the life of the IPL (PROMPT=NO) or changeable via the SET TIME command.

   ETRMODE=YES If a z/OS system is IPLed with ETRMODE=YES, then the PROMPT value is automatically taken as 'NO' and the SET TIME command is not supported, and the time is taken from the LTOD. This system is said to be in *ETR synchronisation mode,* and this system's LTOD will be kept synchronised with the 9037 from here on.

6) In addition, if a 9037 is attached to a 9672, the SE BOC is re-synchronised with the TOD clock at 23:00 each day.

7) And the HMC clock is then reset to the BOC value at 23:15 each day.

Clocks



Time for an explanation!
(Sequence from initial power on)

1) Sysplex timer set initially, from the 9037 console or External Time Source
2) Support Element Battery Operated Clock set initially from HMC
3) If 9037 attached, SE BOC updated from 9037
4) CPC physical TOD clock set initially from SE BOC
5) PR/SM maintains a logical TOD clock, set from CPC TOD when LPAR activated, for each LPAR

When z/OS IPLs, CLOCKnn checked

ETRMODE=NO?

- Use LTOD or issue "SET CLOCK"
- LTOD not synchronised with 9037

ETRMODE=YES?

- LTOD synchronised with 9037
- System now in ETR synchronisation mode, 9037 will maintain synchronisation from here on

1) SE BOC clock reset to TOD at 23:00 daily
2) HMC clock reset to BOC at 23:15 daily

MPOR - 05 - 38

1.33 ETR / TOD synchronisation

The synchronisation process is largely controlled by pure hardware, i.e. the 9037 and the CPC itself. z/OS is only involved if a time discrepancy is detected. There are three different signals sent from the 9037.

1) An **Oscillator** signal is sent initially. This ensures that the CPC PTOD (and therefore the LPAR LTODs), SE BOC and 9037 are all stepped at the same rate. This means that all the physical clocks 'run at the same speed'.

2) **Data Signals** are the actual time and timezone offset signals, along with various 9037 status data, and these are sent at very frequent intervals, every few microseconds (millionths of a second). Timezone data is sent as well as actual time due to potential adjustments such as summer/winter time changes. These data signals are simply stored by the CPC itself.

3) An **On Time Event (OTE)** signal is sent every 1.048567 seconds (every 'mega-microsecond', 220 microseconds, not every million microseconds). This is the actual reference time signal that will be compared with the current PTOD value.

   The TOD clock itself is basically a microsecond counter. It is a 64 bit counter where bit 51 is incremented every microsecond. Bit 32 is known as the 'carry bit' and represents each mega-microsecond. In other words, bit 32 of the TOD is incremented every 1.048567 seconds. The TOD clock in the later generations of the 9672 is actually 128 bits long, to allow for higher resolution timestamps and support sysplex wide unique timestamps (some of the extra bits contain system ids).

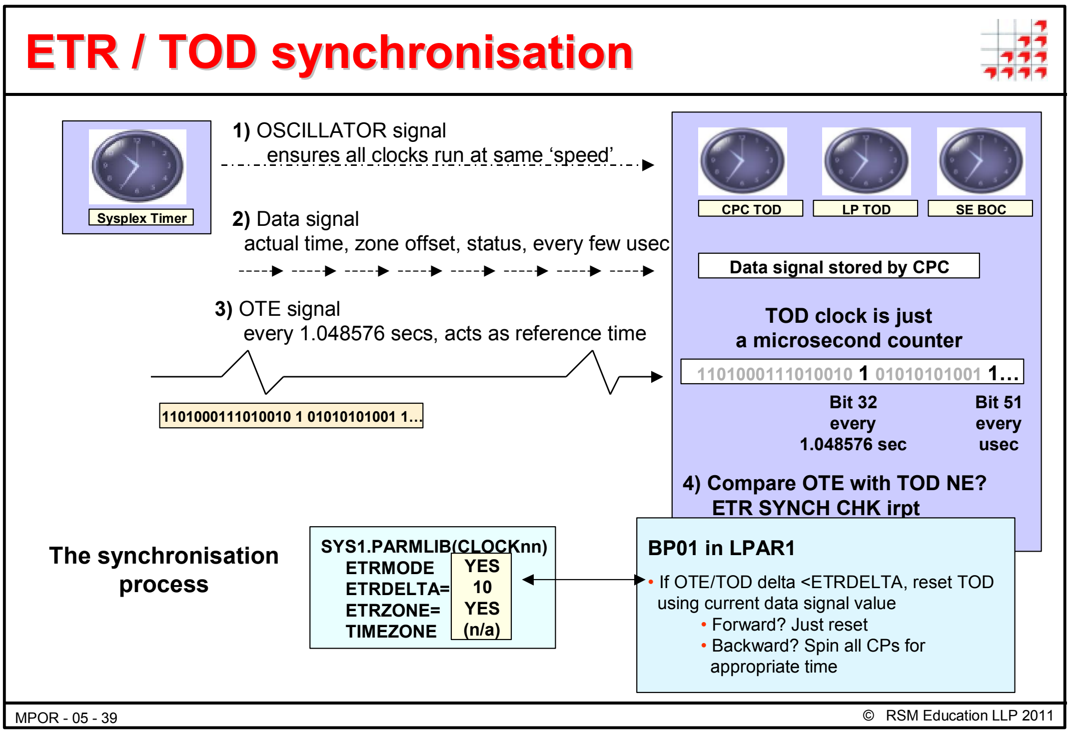
4) So when each OTE signal arrives at the CPC, the CPC compares it with the PTOD value. If the two time values are the same, nothing happens, but if they are different, an **ETR SYNCH CHK interrupt** is issued to any z/OS system running in ETR synch mode.

ETR SYNCH CHK processing

If z/OS is invoked because the TOD time is different from the OTE time value, it is passed the amount of actual discrepancy (delta) and compares this with the **ETRDELTA** value in it's CLOCKnn member. **As long as the delta is less than the ETRDELTA value, z/OS will simply adjust its TOD:**

- If the TOD needs to be set **forward, z/O**S just goes ahead and adjusts the TOD using the current **data signal** value stored by the CPC, and there will be a 'gap' in the timestamps on this system.

- If the TOD needs to be set **backwards,** z/OS can't just set the TOD, as this could cause duplicate or out of sequence timestamps, so z/OS spins all the CPs for the appropriate amount until the TOD has 'caught up' with the correct data signal time.


ETR / TOD synchronisation
Sysplex Timer
1) OSCILLATOR signal
ensures all clocks run at same 'speed'
2) Data signal
actual time, zone offset, status, every few usec
3) OTE signal
every 1.048576 secs, acts as reference time
11010001110100101 01010101001 1...
CPC TOD
LP TOD
SE BOC
Data signal stored by CPC
TOD clock is just
a microsecond counter
1101000111010010 1 01010101001 1...
Bit 32
every
1.048576 sec
Bit 51
every
usec
4) Compare OTE with TOD NE?
ETR SYNCH CHK irpt
The synchronisation
process
SYS1.PARMLIB(CLOCKnn)
ETRMODE YES
ETRDELTA= 10
ETRZONE= YES
TIMEZONE (n/a)
BP01 in LPAR1
• If OTE/TOD delta <ETRDELTA, reset TOD using current data signal value
• Forward? Just reset
• Backward? Spin all CPs for appropriate time
MPOR - 05 - 39
© RSM Education LLP 2011

1.34 ETRDELTA

Before we look at what happens if the TOD clock 'drifts' by more than the ETRDELTA value, we need to look at the implications of this environment and consider what the ETRDELTA value really means.

1) First, the fact that all the clocks are being stepped at the same rate means that there is no real reason for, and very little likelihood of, them being out of step, and therefore producing a different time value.
   However, we still have to allow for the fact that different physical oscillators are being used in the two separate clocks so discrepancies could occur.

2) The data signals are sent at **very** frequent intervals (every few microseconds).

3) If an adjustment is made by z/OS because of an ETR SYNCH CHK, it will normally be a **very small** correction.

This all adds up to the fact that, unless something goes seriously wrong, the two clocks should always be synchronised to within a few microseconds.

So how could a discrepancy occur?

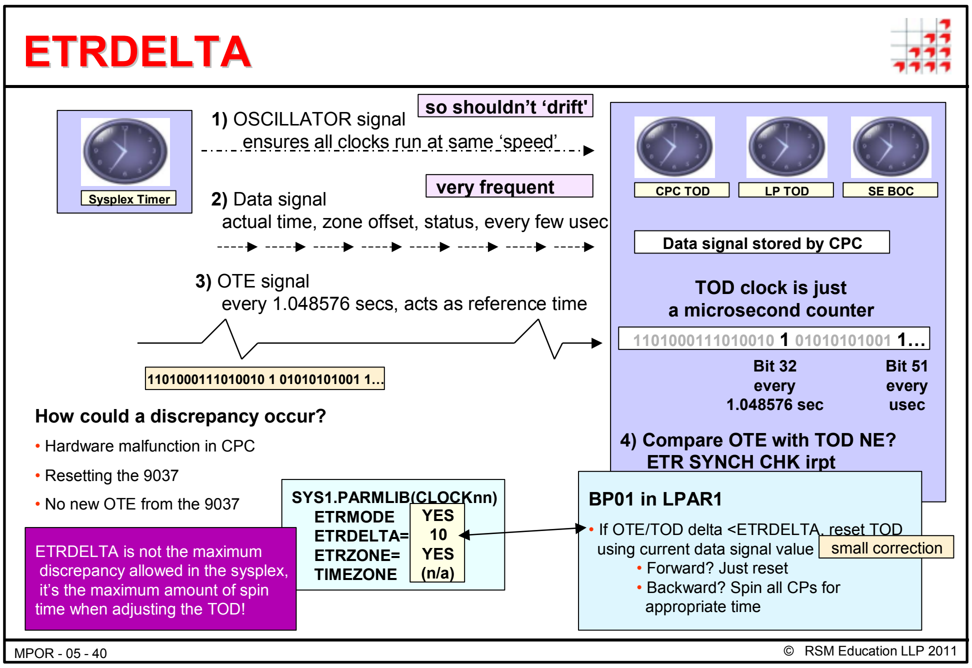
As mentioned above, very small discrepancies could occur due to the fact that we're dealing with separate physical oscillator functions. Apart from that though there are only a few reasons:

- there could have been a hardware malfunction in the CPC or 9037, resulting in either the TOD or 9037 moving wildly away from its previous value

- it may be that someone has reset the values on the 9037 while the sysplex was running (what we call in the trade a non-career enhancing move). The TOD can't be reset manually from z/OS if that z/OS is running in ETR synch mode, so the discrepancy must have been caused by a 9037 adjustment

- or most likely, when the CPC checks the OTE signal at TOD 'carry time', the OTE is unchanged, because no new OTE has arrived from the 9037 - **because this system has lost its connection to the 9037.**

It is extremely unlikely that the ETRDELTA value would be exceeded in normal operations.



So what about this ETRDELTA value?

Everything previously described means that unless the system has lost its connection to the 9037, its TOD will normally not drift by more than a few microseconds. So what value should you code for ETRDELTA?

Firstly, you need to realise that using the default ETRDELTA of 10 seconds **does not mean you will allow timestamp discrepancies of almost 20 seconds in the sysplex** (one machine just under 10 seconds slow, one just under 10 seconds fast). The maximum discrepancy for any TOD clock is in effect 1.048576 seconds, as that is the frequency of the OTE signals. The real meaning of the ETRDELTA value is twofold -

- it represents the theoretical maximum amount of spin time if the TOD need to be set backwards, but even this is likely to be limited to 1.048567 seconds, and
- if you attach a 9037 to a running system, an OTE synch check is performed against that system in the same way as described previously. If the ETRDELTA is exceeded here, that system is forced out of ETR synch mode. It is much more unlikely that this process would work if a very low ETRDELTA were specified, as this would mean the 9037 would have to be set very precisely to the target system's TOD before attachment.

All in all, a default ETRDELTA of 10 seconds is probably best left alone.

The mechanics of the synchronisation process mean that once a 9037 is attached, all TODs in the sysplex **will** be synchronised to within a few microseconds, **regardless of the ETRDELTA value specified.**

Timer problems are far more likey to be connectivity or actual 9037 problems

1.35 Sysplex Timer connectivity problems

If a machine detects that it has lost its connection to a sysplex timer, it will invoke z/OS and raise a ***"lost connectivity to the sysplex timer"*** condition.

z/OS will then suspend operation and issue the following rather wordy message -

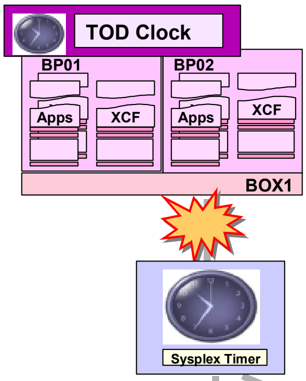
```
IEA015A THIS SYSTEM HAS LOST ALL CONNECTION
TO THE SYSPLEX TIMER

IF THIS EVENT OCCURRED ON SOME, BUT NOT ALL
SYSPLEX MEMBERS, THE LIKELY CAUSE IS A LINK
FAILURE. TO FIX, ENSURE THAT EACH AFFECTED
SYSTEM HAS AT LEAST ONE CORRECTLY
CONNECTED AND FUNCTIONAL LINK.

IF THIS EVENT OCCURRED ON ALL SYSPLEX
MEMBERS, THEN THE LIKELY CAUSE IS A SYSPLEX
TIMER FAILURE. TO FIX, REFER TO THE MESSAGE
IEA015A DESCRIPTION IN MVS SYSTEM MESSAGES.

AFTER FIXING THE PROBLEM, REPLY "RETRY" FROM
THE SERVICE CONSOLE(HMC).IF THE PROBLEM WAS
NOT CORRECTED, THIS MESSAGE WILL BE REISSUED
AND YOU MAY TRY AGAIN. REPLY "ABORT" TO EXIT
MESSAGE LOOP. PROBABLE RESULT: 0A2-114
WAITSTATE.
```

Well, it may be wordy, but it is pretty self explanatory!

If you can fix the problem, do so and reply **RETRY**. You should then be capable of resuming the affected systems.

If you reply **ABORT**, then message IXC462W is issued and the affected systems are placed into the 0A2-114 wait state and must be re-IPLed. However, they can't re-IPL into the sysplex until the sysplex timer connection is re-established correctly.

Sysplex timer connectivity problems



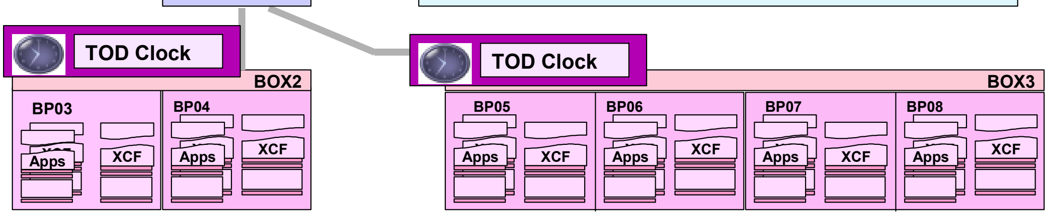
If the sysplex timer is lost

- On the affected systems

```
IEA015A  THE SYSTEM HAS LOST ALL CONNECTIONS TO THE SYSPLEAX MTIMER
         (plus lots of other text – see notes for actual text)
         (reply RETRY or ABORT)
```

- Can fix the connection?
  - Do so, reply RETRY and should be ok
  - If connection not fixed, message repeats
- Can't fix the connection?
  - Reply ABORT, will get

```
IXC462W XCF IS UNABLE TO ACCESS THE ETR AND HAS PLACED THIS SYSTEM
        INTO NON-RESTARTABLE WAIT STATE CODE: 0A2 REASON CODE: 114
```



MPOR - 05 - 41

1.36 Losing the sysplex timer

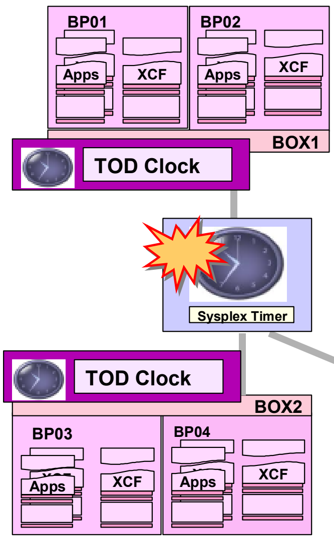
If you lose the sysplex timer altogether (and don't have a backup), then **all systems** are placed in the IEA015A message loop.

The options are the same as on the previous page, but of course the potential impact is even more serious. If the problem can't be fixed, the entire sysplex will be lost, and cannot be restablished.

It would be possible to bring up a temporary sysplex configuration, for example all systems on BOX3, by using the **SIMETRID** parameter in CLOCKnn. Remember that TOD clocks are machine, not LPAR based.

SIMETRID is a two-digit value indicating a simulated ETR. If all systems on BOX3 were re-IPLed with the same SIMETRID value, they could form a valid multisystem sysplex, but would still have to be shut down and re-IPLed again when the real sysplex timer was fixed.

Losing the sysplex timer



If the last or sysplex timer itself fails

- On ***all*** systems

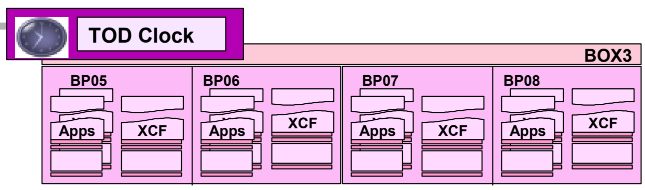
```
IEA015A  THE SYSTEM HAS LOST ALL CONNECTIONS TO THE SYSPLEX TIMER
         (plus lots of other text - see notes for actual text)
         (reply RETRY or ABORT)
```

- Same as before, it can't fix the problem, reply ABORT, will get

```
IXC462W XCF IS UNABLE TO ACCESS THE ETR AND HAS PLACED THIS SYSTEM
        INTO NON-RESTARTABLE WAIT STATE CODE: 0A2 REASON CODE: 114
```

- Can't re-IPL the sysplex until sysplex timer problem resolved

You could bring up, for example, a 4-way sysplex (BOX3), using 'SIMETRID'



MPOR - 05 - 42

1.37 Couple Data Set problems

In all the examples where we've discussed the various Couple Data Sets, we've suggested that you should definitely define an alternate and should consider having a spare. The reason for this is that if you don't have an alternate, losing a Couple Data Set can have disastrous consequences.

SYSPLEX	If you lose this, you lose all systems. End of story.
CFRM	Lose this and you lose all systems as well
SFM	Lose this and you'll carry on running ok, SFM won't be around to handle your connectivity failures and status update missing conditions. You'll have to rely on manual reconfiguration procedures
WLM	If you lose the WLM data set, all systems currently in goal mode will continue in goal mode, but any changes to the WLM policy will not be picked up by these systems. Also each goal mode system is now running independently and can't participate in any form of workload balancing.
ARM	If you lose this, you will have no ARM policy in place and so no restart management will be available
LOGR	Lose this and the System Logger services will cease to function. The effects of this vary from inconvenient (OPERLOG) to extremely nasty (CICS).

Couple Data Set problems

Sysplex Couple Data Set — **Lose this, all systems load 0A2 wait state**

Policy 1 CFRM Couple Data Set — **Lose this, all systems also load 0A2 wait state**

Losing access to (or just losing) a Couple Data Set without an alternate -
If either of the above, you (or all systems) are placed into non-restartable 0A2

- *Check any messages involving Couple Data Sets very carefully*
- *Other Couple Data Sets (below) involve loss of facility, rather than loss of systems*

Use alternates!

Policy 1 WLM Couple Data Set — **Lose this, and each WLM stays in Goal mode, but runs independently**

Policy 1 ARM Couple Data Set — **Lose this, and - you're ARMless**

Policy 1 SFM Couple Data Set — **Lose this, and the systems continue without an active SFM policy**

Policy 1 LOGR Couple Data Set — **Lose this, and you lose access to System Logger services**

MPOR - 05 - 43

Changing COUPLE parameters

```
RO RSMB,SETXCF COUPLE,INTERVAL=20
RO RSMB,SETXCF COUPLE,OPNOTIFY=23
```

MPOR - 05 - 44

1.38 Failures in the Coupling Facility environment

The next major area to look at is failures in the Coupling Facility environment.

In this section we're not thinking about signalling failures, just Coupling Facility environment failures and how they will affect applications like IMS, DB2, etc.

Coupling Facility failure

- If you lose the only Coupling Facility, then all applications will lose access to their structures. Some applications, like JES, can survive this but most product applications like IMS, DB2, etc. can't

- If you lose a Coupling Facility but have another one, it is a case of rebuilding the lost structures into the alternate facilities.

CFC failure

- If you lose the only connection from a z/OS system to a Coupling Facility, for that system it is as if you've lost the Coupling Facility itself, so the situation is the same as above.

- If you lose a CFC from one system but have another one in use, it is just a capacity problem.

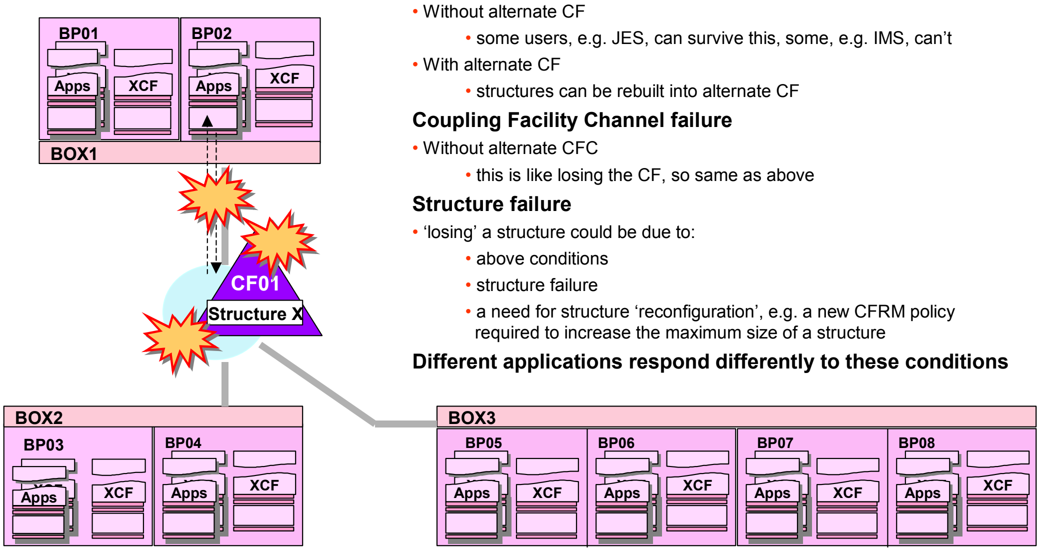
Structure failure

- Structure 'failure' could be due to any of the above conditions, or it could be an actual failure (storage failure, etc.) of the structure itself.

- It could be caused by a need for structure 'reconfiguration', for example because the CFRM policy needs to be updated to allow a new maximum size for a structure.

In all cases of structure failure, we're talking about the need to rebuild a structure, either within the current facility or an alternate.

Failures in the Coupling Facility environment



Coupling Facility failure

- Without alternate CF
  - some users, e.g. JES, can survive this, some, e.g. IMS, can't
- With alternate CF
  - structures can be rebuilt into alternate CF

Coupling Facility Channel failure

- Without alternate CFC
  - this is like losing the CF, so same as above

Structure failure

- 'losing' a structure could be due to:
  - above conditions
  - structure failure
  - a need for structure 'reconfiguration', e.g. a new CFRM policy required to increase the maximum size of a structure

Different applications respond differently to these conditions

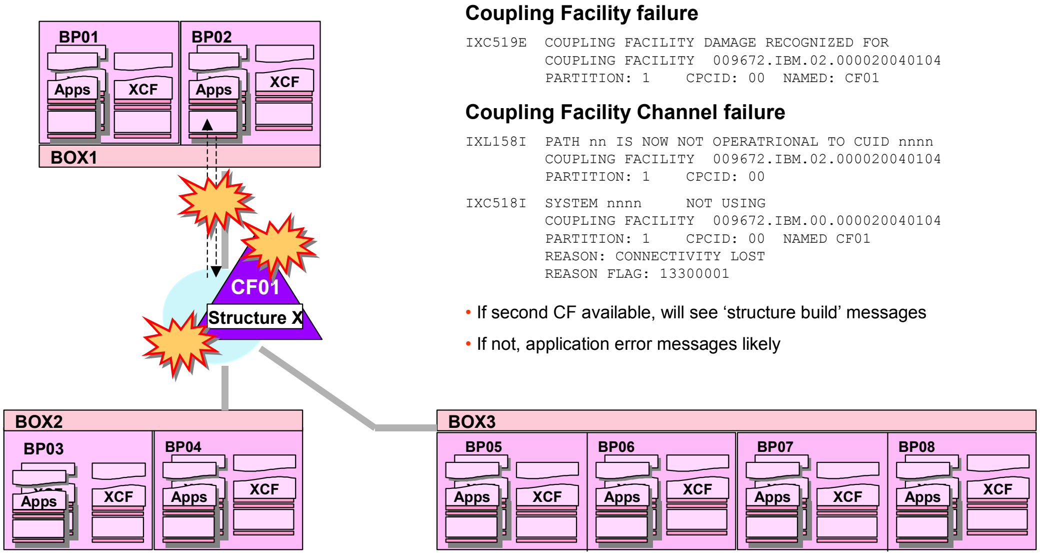
MPOR - 05 - 45

1.39 Coupling Facility and CFC error indicators

Shown opposite are the sorts of messages you will see in the event of either Coupling Facility or CFC failure.

If alternates are available, you will probably also see various messages related to structure rebuild and if not, probably a whole bunch of application failure messages.

Coupling Facility & CFC error indicators



Coupling Facility failure

```
IXC519E  COUPLING FACILITY DAMAGE RECOGNIZED FOR
         COUPLING FACILITY  009672.IBM.02.000020040104
         PARTITION: 1    CPCID: 00   NAMED: CF01
```

Coupling Facility Channel failure

```
IXL158I  PATH nn IS NOW NOT OPERATRIONAL TO CUID nnnn
         COUPLING FACILITY  009672.IBM.02.000020040104
         PARTITION: 1    CPCID: 00

IXC518I  SYSTEM nnnn     NOT USING
         COUPLING FACILITY  009672.IBM.00.000020040104
         PARTITION: 1    CPCID: 00   NAMED CF01
         REASON: CONNECTIVITY LOST
         REASON FLAG: 13300001
```

- If second CF available, will see ‘structure build’ messages
- If not, application error messages likely

MPOR - 05 - 46

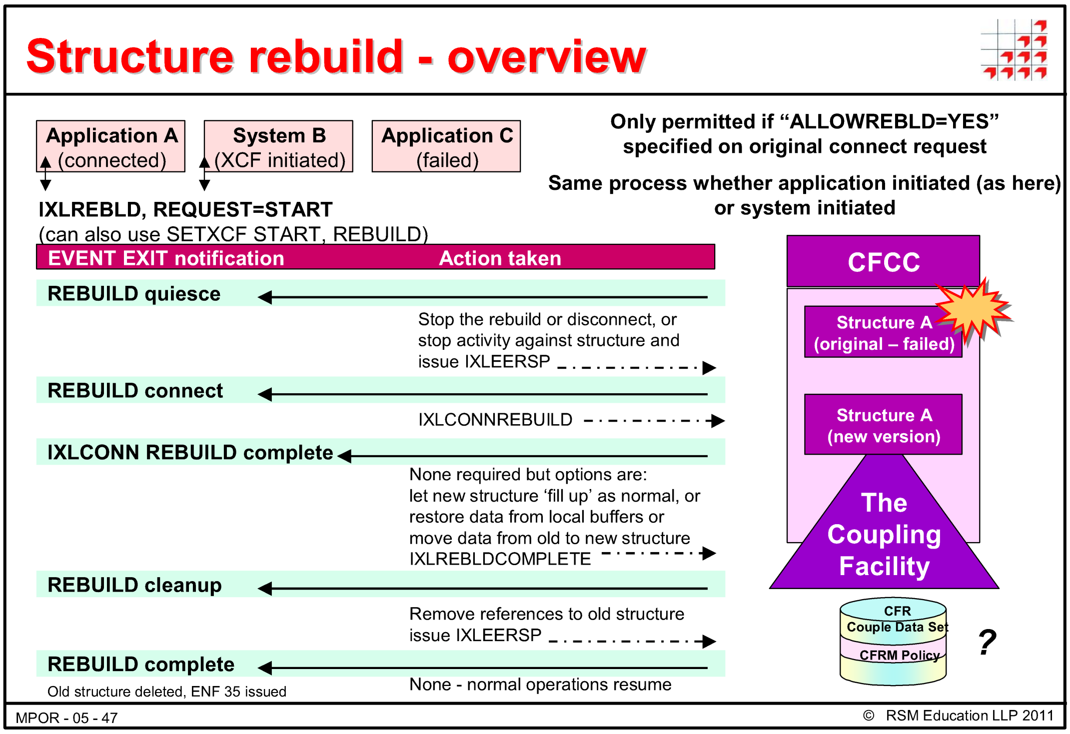
1.40 Structure rebuild - an overview

If you've not yet attended RSM's **Parallel Sysplex Internals & Fundamentals** course, you might not be aware of the effort involved in structure rebuild.

In which case, don't worry about the detail of the picture opposite. It is making the point that when a structure rebuild is requested, all the application instances that are connected to that structure have to get involved with the process. There's an awful lot of activity involved and it's all done by committee decision! Everybody has to approve each stage of the process.

The more systems, and connections, there are in your sysplex, the more complicated it all gets.

The good news is that almost all of this activity is internal, and there's very little operator involvement with the actual process itself.


Structure rebuild - overview
Application A (connected)
System B (XCF initiated)
Application C (failed)
Only permitted if "ALLOWREBLD=YES" specified on original connect request
Same process whether application initiated (as here) or system initiated
IXLREBLD, REQUEST=START
(can also use SETXCF START, REBUILD)
EVENT EXIT notification
Action taken
REBUILD quiesce
Stop the rebuild or disconnect, or stop activity against structure and issue IXLEERSP
REBUILD connect
IXLCONNREBUILD
IXLCONN REBUILD complete
None required but options are: let new structure 'fill up' as normal, or restore data from local buffers or move data from old to new structure
IXLREBLDCOMPLETE
REBUILD cleanup
Remove references to old structure issue IXLEERSP
REBUILD complete
Old structure deleted, ENF 35 issued
None - normal operations resume
CFCC
Structure A (original – failed)
Structure A (new version)
The Coupling Facility
CFR
Couple Data Set
CFRM Policy
?
MPOR - 05 - 47
© RSM Education LLP 2011

1.41 Structure rebuild - why?

Structure rebuild is the process of recovering data lost in the Coupling Facility environment in the event of the sorts of failures mention a few pages back.

Remember, this data is only actually *a copy of data already held in the applications' local buffers,* and so in the event of structure loss, it can be rebuilt from those local buffers.

Structure loss, and therefore structure rebuild, can occur because of three conditions.

1) actual **structure damage,** i.e. a failure within a Coupling Facility (a bit like an I/O error on DASD).

2) **Coupling Facility failure,** perhaps a hardware failure for example.

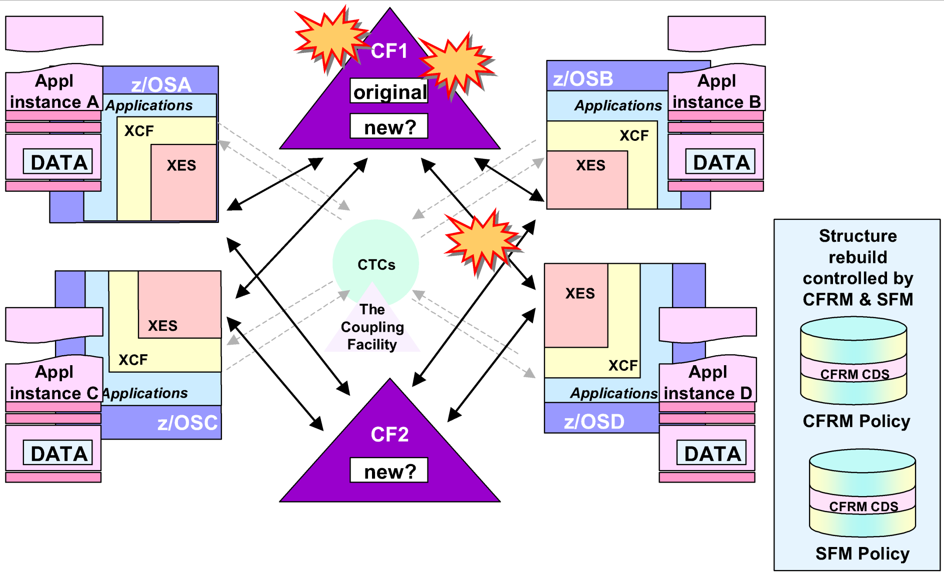
3) or a **loss of connectivity to a Coupling Facility.** If a coupling link is lost from a system to a Coupling Facility, for all applications connected to structures in that CF on that system it's as if the structures or the Coupling Facility itself has failed.

In the first case above, the structure could simply be rebuilt in the same Coupling Facility, and in the second two cases, it could be rebuilt into an alternate Coupling Facility.

Parameters in both the CFRM and SFM policies control the rebuild process.

This is the reason you should use a standalone CF for production. If you lose a CF LPAR and a z/OS LPAR because the machine has died, you have lost not just the structures in the CF but also some of the local buffers needed to rebuild the data.

Structure rebuild – why?



MPOR - 05 - 48

1.42 Structure rebuild controls

Structure rebuild might be initiated automatically in the event of a CFC failure. This will happen if there is an active SFM policy with **CONNFAIL(YES)** coded, and it is the combination of:

- the **REBUILDPERCENT** in the **CFRM policy** and
- the **SYSTEM WEIGHT** in the **SFM policy**

This determines if the system will initiate rebuild processing for a structure or structures.

The calculation is as follows.

1) Assume Str 01 has a rebuild percentage of 50% (C)

2) System A loses its CFC connection to CF01

3) 'A' is "the combined system weights of all systems that have lost connectivity to the Coupling Facility and on which there exists a user of a structure in that facility"
In this example, this means System A, which has a weight of 10

4) 'B' is "the combined system weights of all systems on which there exists a user of a structure in the facility to which connectivity has been lost"
In this example, this means Systems A, B, C & D, which have a combined weight of 13, because System A has an explicit weight of 10 and the others have the default weight of 1.

5) so the SFM performs the following calculation -

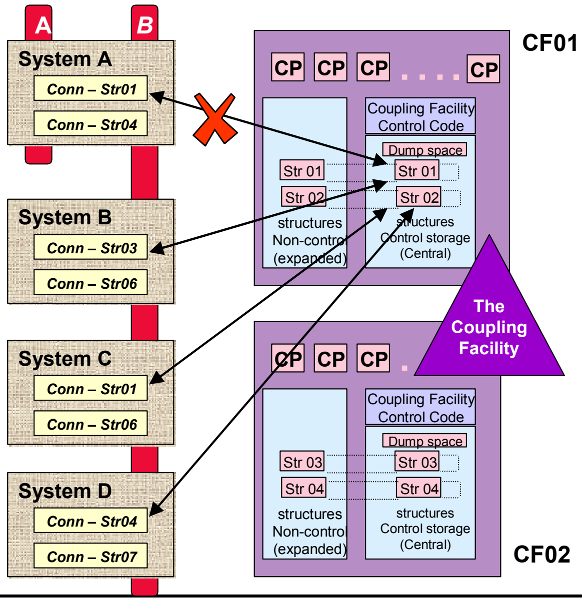
   is (A / B * 100) ge C?

   is (10 / 13 * 100) ge 50?

   is 76.92 ge 50?

If the answer is "**yes**" then the system will initiate **rebuild processing** for the structure, and that is the case here.

Structure rebuild controls



CFRM Policy C

STRUCTURE NAME(STR01) REBUILDPERCENT(50)

SFM Policy

CONFAIL(YES)
SYSTEM NAME(*) WEIGHT(1)

SYSTEM NAME(SystemA) WEIGHT(10)

Rebuild calculation

- System A loses connection to CF01
- A = weight of systems that have lost connectivity
- B = weight of all systems with connections to CF01
- The calculation is

  is A / B * 100 ge C?
- If yes – rebuild structure

MPOR - 05 - 49

1.43 Structure rebuild - application support

As mentioned a couple of pages ago, not all applications support structure rebuild. The chart opposite shows you who does and who doesn't. Some points:

DB2 DB2 now the rebuild process and is automatic.

JES2/3 JES should be set up to duplex a copy of its structure on DASD and have a 'standby' structure available as well. Switching to the standby can be either automatic or invoked via the JES reconfiguration dialog.

In addition, **XCF, VTAM** and the **System Logger** support rebuild, but ignore the REBUILDPERCENT parameter. With these three, structures are always rebuilt if **any** connectivity is lost.

Structure rebuild – applications support

Application	Structure	Rebuild allowed?	'rebuildpercent' supported?	Comments
IRLM	IMS lock structure	Yes	Yes	
IMS	OSAM cache structure	Yes	Yes	
IMS	VSAM cache structure	Yes	Yes	
IRLM	DB2 lock structure	Yes	Yes	
DB2	GBP cache structure	Yes	Yes	
DB2	SCA list structure	Yes	Yes	
SMSVSAM	lock structure	Yes	Yes	
SMSVSAM	VSAM cache structures	Yes	Yes	
JES2/3	CHKPT list structure	No	No	Checkpoint reconfiguring dialog
RACF	cache structures	Yes	Yes	
System Logger	Logstream list structures	Yes	No	Rebuilt is any connectivity loss
GRS	STAR lock structure	Yes	Yes	
XCF	signalling list structure	Yes	No	Rebuilt is any connectivity loss
VTAM	generic resources structure	Yes	No	Rebuilt is any connectivity loss

MPOR - 05 - 50

1.44 Automatic Restart Manager

So far, we've been looking at problems related to systems, paths, Coupling Facilities, etc. Now, we're going to look at what the **Automatic Restart Manager** can do for us. ARM provides the ability to restart work subsystem address spaces like VTAM, CICS, DB2, etc, whether they're running as batch or started tasks.

The Automatic Restart Manager:

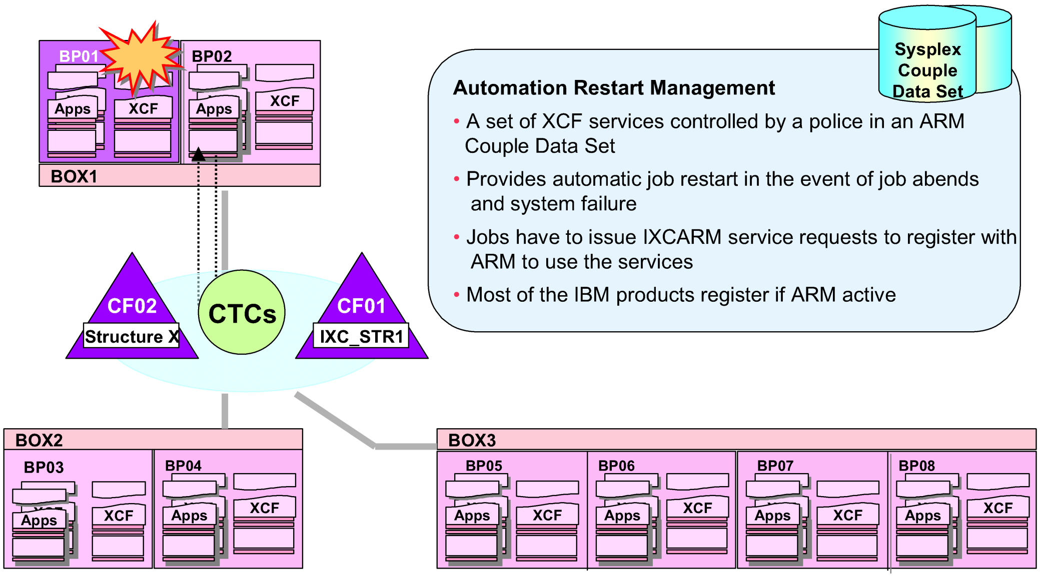
- restarts failed batch jobs or started tasks after a system or job failure
- supports job inter-dependencies on the restarts

Although it will support batch jobs, what we're really talking about here is the ability to restart subsystem products rather than your general batch workload:

- if the **application** fails, it will be restarted on the same system
- if a **system** fails, its applications will be started on a different system

ARM is controlled through an ARM policy in an ARM Couple Data Set, but there is an additional step involved here. Programs wishing to use ARM services must also **register** with ARM via the **IXCARM** service macro. This means that programs have to be **coded** to use ARM. The newer releases of the IBM products like CICS, IMS etc do this. If you set up the ARM environment for them, these products will be automatically restarted in the event of the failures described above.

Automatic Restart Manager



Automation Restart Management

- A set of XCF services controlled by a police in an ARM Couple Data Set
- Provides automatic job restart in the event of job abends and system failure
- Jobs have to issue IXCARM service requests to register with ARM to use the services
- Most of the IBM products register if ARM active

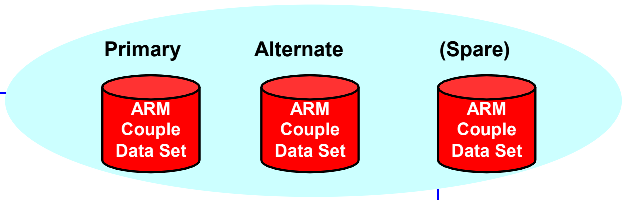
MPOR - 05 - 51

1.45 The ARM policy

The ARM policy parameters work as follows.

RESTART_ ORDER	This section defines the order in which elements will be restarted
LEVEL	Specifies a level (order) from 2 to 65536. Level 1 is reserved
ELEMENT_NAME	The names of the elements on this level. The name used by the program to register with ARM is not necessarily the jobname or STC name
ELEMENT_TYPE	A generic name that an element can use to register instead of its actual element name
RESTART_GROUP	The name of a group of elements to be restarted together in the event of a cross-system restart. These names are installation defined
TARGET_SYSTEM	The systems, in order of preference, to be used for cross-system restarts
FREE_CSA	How much free CSA (below/above) required on a target system before this system chosen for cross-system restarts
RESTART_PACING	Wait this long between restarts when restarting multiple elements on a target system
ELEMENT	The name of the element. The following parameters define the restart requirements for this element. Element names are not the same as jobnames/STC names
RESTART_ATTEMPTS	How many attempts at restart (max 3) over what period. This stops ARM recursively restarting an element that keeps failing.
RESTART_TIMEOUT	How long to wait after restarting an element for the subsequent REGISTER request before considering this a failed restart
READY_TIMOUT	How long to wait after the REGISTER request for the subsequent READY request before considering this a failed restart
TERMTYPE	ELEMTERM means restart only in the event of abend, ALLTERM means restart in the event of either abend or system failure
RESTART_METHOD	The JCL or START command text to be used for restarting this element. Consists of the following two parameters:
“if this type of error”	
ELEMTERM	means if the restart is due to the element abending, use the values in the “restart this way” for restart, otherwise use the original JCL/command
SYSTERM	means if the restart is due to the system failing, use the values in the “restart this way” for restart, otherwise use the original JCL/command
BOTH	means use the values in the “restart this way” for the restart, regardless of the reason
“restart this way”	
PERSIST	means use the original JCL or start command text
JOB,’dsname(mbr)’	is the specific JCL to use for the restart
STC,’text’	is the actual start command to use for the restart

The ARM policy



ARM policy options

```
RESTART_ORDER
    (LEVEL(2)
        ELEMENT_NAME(sys??pay*,abcjob)

    LEVEL(3)
        ELEMENT_NAME(xyztask)

RESTART_GROUP(                          name of group)
    TARGET_SYSTEM(                      SystemA,SystemB)
    FREE_CSA(                           below,above)
    RESTART_PACING(                     delay in secs between restarts in grp)
        ELEMENT(                        element name)
            RESTART_ATTEMPTS(           max #, over what period)
            RESTART_TIMEOUT(            interval between restart and REGISTER)
            READY_TIMEOUT(              interval between REGISTER and READY)
            TERMTYPE(                   ALLTERM or ELEMTERM)
            RESTART_METHOD(             "if this type of error", "restart this way")
        ELEMENT
            (restart parameters for next element)
```

Batch Job or Started Task

IXCARM REQUEST=REGISTER, Element=name

MPOR - 05 - 52

1.46 The ARM defaults

You can start using ARM without actually setting up a policy. In this way, you will be using the ARM defaults, as shown opposite. This is not a good idea as the defaults are non-specific but different subsystems will need specific values.

The defaults are that DB2, IMS and VTAM are all associated with RESTART LEVEL 1, and CICS and CICSplex are associated with RESTART LEVEL 2, along with everything else. Also, everything will be eligible for restart in all circumstances and cross-system restarts are randomly targeted.

This environment won't work in practice, so you'll need to set up a policy to provide specifics for your installation's particular needs. You'll probably want to implement ARM one environment at a time, and so you should create a policy which includes the default RESTART_GROUP but with the RESTART_ATTEMPTS value set to zero. This will turn ARM off for everything by default; then you can code specific groups for the subsystems you want to use the service.

Activating the environment is done in the usual way:

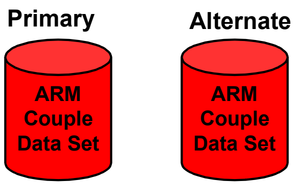
```
SETXCF COUPLE,TYPE=ARM,PCOUPLE=dsn
```

```
SETXCF COUPLE,TYPE=ARM,ACOUPLE=dsn
```

And then activate the policy by:

```
SETXCF START,POLICY,TYPE=ARM,POLNAME=name
```

The ARM defaults



Beware the defaults

- *Although you can activate ARM and use the defaults, you should not do so*
- *The defaults are effectively random and won;t necessarily work for individual applications like CICS, DB2, etc.*
- *In any policy you create, to nullify the defaults, include*
  RESTART_GROUP(DEFAULT)
  ELEMENT(*)
  RESTART_ATTEMPTS(0,300)
- *Code explicit group/element statements for the work you actually want covered*

ARM defaults

RESTART_ORDER
*(LEVEL(**1**)*
DB2, IMS, VTAM always restarted first)
*LEVEL(**2**)*
*ELEMENT_TYPE(**SYSLVL2**)*

*RESTART_GROUP(**DEFAULT**)*
TARGET_SYSTEM(*****)
*FREE_CSA(**0,0**)*
*RESTART_PACING(**0**)*
ELEMENT(*)
*RESTART_ATTEMPTS(**3,300**)*
*RESTART_TIMEOUT(**300**)*
*READY_TIMEOUT(**300**)*
*TERMTYPE(**ALLTERM**)*
*RESTART_METHOD(**BOTH,PERSIST**)*

MPOR - 05 - 53

1.47 Manipulating the ARM environment

Shown opposite are some of the commands to display and activate the ARM environment.

Manipulating the ARM environment

Activating the defaults (bad idea)

```
DISPLAY XCF,POLICY,TYPE=ARM
IXC364I  00.25.03 DISPLAY XCF
         TYPE: ARM
         POLICY NOT STARTED
```

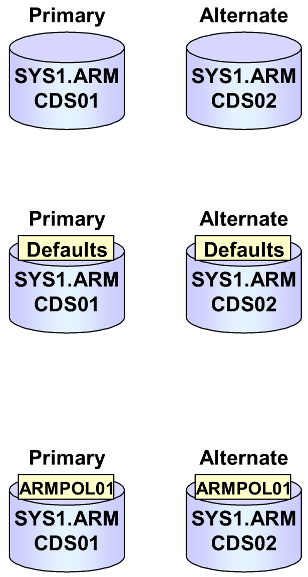
```
SETXCF START,POLICY,TYPE=ARM
IXC805I  ARM POLICY HAS BEEN STARTED BY SYSTEM BP01
         POLICY DEFAULTS ARE NOW IN EFFECT
```

```
DISPLAY XCF,POLICY,TYPE=ARM
IXC364I  00.27.22 DISPLAY XCF
         TYPE: ARM
         POLNAME:        POLICY DEFAULTS ARE IN EFFECT
         STARTED:        05/30/09 00.26.12
         LAST UPDATED:   -- --
```

Activating an installation defined policy

```
SETXCF START,POLICY,TYPE=AARM,POLNAME=ARMPOL01
IXC8051  ARM POLICY HAS BEEN STARTED BY SYSTEM BP01
         POLICY NAMED ARMPOL01 IS NOW IN EFFECT
```

```
DISPLAY XCF,POLICY,TYPE=ARM
IXC364I  00.35.25 DISPLAY XCF
         TYPE: ARM
         POLNAME:        ARMPOL01
         STARTED:        05/30/09 00.26.12
         LAST UPDATED:   05/30/09 00.34.53
```



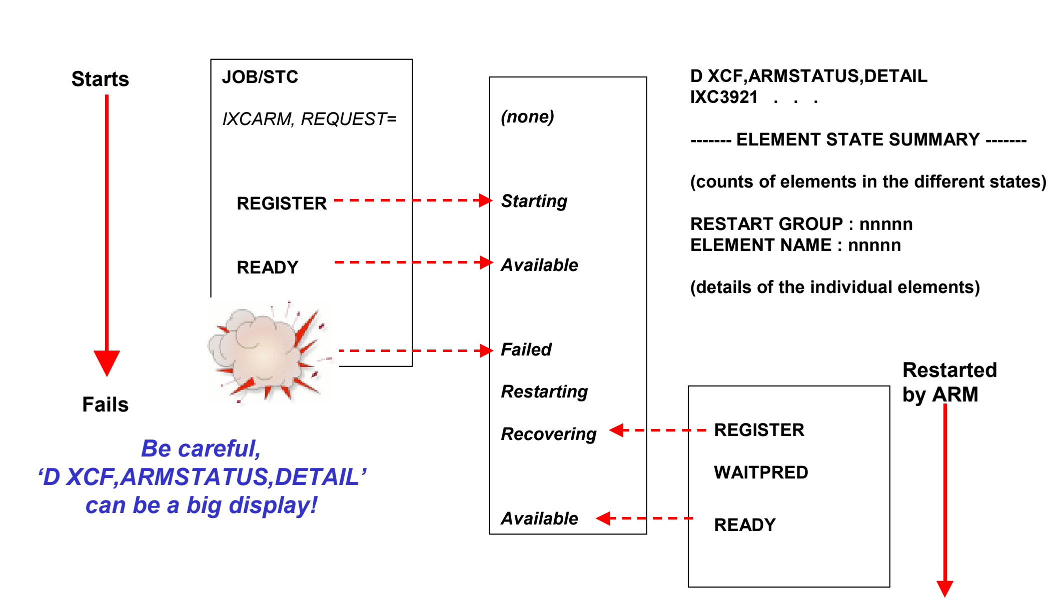
MPOR - 05 - 54

1.48 ARM element states

We said the programs have to **register** with ARM via the IXCARM service. There are actually several different **element states** that are noted for a job, depending on where it is in the ARM process.

These states are shown opposite, and can be displayed on the console via the **D XCF,ARMSTATUS [,DETAIL]** command.

ARM element states



MPOR - 05 - 55

1.49 D XCF ARMSTATUS

If we issue this command initially, we see there on no jobs currently registered with ARM.

Job BEPEJOB1 then starts, and registers with ARM.

Note, there are no standard IXC messages issued as the IXCARM service is invoked. If a program chooses to, it could issue WTOs, but it's not a requirement.

If we then enter the **D XCF,ARMSTATUS,DETAIL** command we can see the data for the newly registered job. Points to note:

- the **Element State Summary** shows the number of elements (jobs) in the various ARM states
- for each element registered, you then get its status
- you can see that BEPEJOB1 is part of the **default** restart group, and if you compare the various parameters shown here with the defaults a few pages back you can see we're using the ARM defaults on this system
- the current state of this element is **AVAILABLE**
- the **CURR SYS** and **INIT SYS** show the current and initial systems for this job. If these are different then this element has been cross-system restarted
- the number of restarts is shown ('0' in this case)
- the date and time of any restarts would also be listed if the element had been restarted.

"D XCF ARMSTATUS"

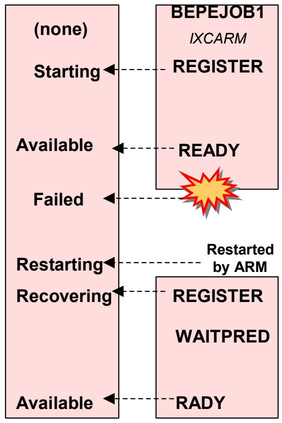
D XCF,ARMSTATUS

```
IXC392I  00.52.12 DISPLAY XCF
         NO ARM ELEMENTS ARE DEFINED

$HASP373 BEPEJOB1 STARTED - INIT A - CLASS F - SYS BP01
IEF493I BEPEJOB1 STARTED - TIME=00.53.14
```

(job registers with ARM)

D XCF,ARMSTATUS,DETAIL

```
IXC392I 00.54.32 DISPLAY XCF
   ARM RESTARTS ARE ENABLED
   ----------------ELEMENT STATE SUMMARY-------- -TOTAL- -MAX-
   STARTING AVAILABLE FAILED RESTARTING RECOVERING
          0         1      0          0          0     1    20
   RESTART GROUP:DEFAULT         PACING  : 0       FREE CSA: 0       0
   ELEMENT NAME :BEPEJOB1        JOBNAME :BEPEJOB1 STATE   :AVAILABLE
     CURR SYS :BP01              JOBTYPE :JOB      ASID     :002D
     INIT SYS :BP01              JESGROUP:BPPLEX01 TERMTYPE:ALLTERM
     EVENTEXIT:GOSSIP99          ELEMENTYPE:*NONE* LEVEL    :   2
     TOTAL RESTARTS :       0    INITIAL START:05/30/09 00.53.14
     RESTART THRESH :0 OFF 3     FIRST RESTART:*NONE*
     RESTART TIMEOUT:     300    LAST RESTART :*NONE*
```



MPOR - 05 - 56

1.50 ARM restart, same system

1) Our batch job starts, and registers with ARM

2) It is then cancelled, but with the **ARMRESTART** option. ARM does not restart jobs that are **cancelled** unless this parameter is added on the cancel command.

The job is cancelled but restarted on the same system, as shown, and re-registers with ARM.

3) If we do a display now, we can see that the element is **AVAILABLE** and has now had one restart at the date and time shown.

ARM restart, same system

(Batch job starts)

```
$HASP373 BEPEJOB1 STARTED - INIT A - CLASS F - SYS BP01
IEF493I  BEPEJOB1 STARTED - TIME=00.53.14
```

(job registers with ARM)

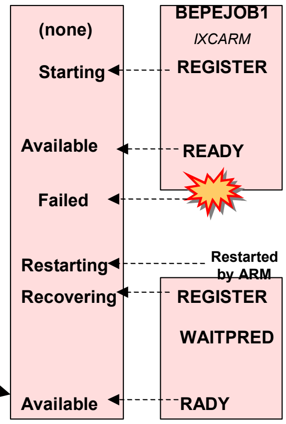
C BEPEJOB1,ARMRESTART

```
IEE301I BEPEJOB1 CANCEL COMMAND ACCEPTED
IXC812I JOBNAME BEPEJOB1, ELEMENT BEPEJOB1 FAILED
        THE ELEMENT WAS RESTARTED WITH PERSISTENT JCL

$HASP373 BEPEJOB1 STARTED - INIT A - CLASS F - SYS BP01
IEF493I  BEPEJOB1 STARTED - TIME=00.56.54
```

(job re-registers with ARM)

D XCF,ARMSTATUS,DETAIL

```
IXC392I 00.58.32 DISPLAY XCF
   ARM RESTARTS ARE ENABLED
   -------------- ELEMENT STATE SUMMARY ---------  TOTAL- -MAX-
   STARTING AVAILABLE FAILED RESTARTING RECOVERING
          0         1      0          0          0      1     20
   RESTART GROUP:DEFAULT        PACING  : 0       FREE CSA:   0    0
   ELEMENT NAME:BEPEJOB1        JOBNAME :BEPEJOB1 STATE   :AVAILABLE
     CURR SYS :BP01             JOBTYPE :JOB      ASID    :007F
     INIT SYS :BP01             JESGROUP:BPPLEX01 TERMTYPE:ALLTERM
     EVENTEXIT:GOSSIP99         ELEMENTYPE:*NONE* LEVEL   :   2
     TOTAL RESTARTS :       1   INITIAL START:05/30/09 00.53.14
     RESTART THRESH : 0 OF 3    FIRST RESTART:05/30/09 00.56.54
     RESTART TIMEOUT:     300   LAST RESTART :05/30/09 00.56.54
```



MPOR - 05 - 57

1.51 ARM restart, cross-system

Continuing from before:

1) Our job is still running, following its restart from the previous page.

   - The system it's running on (BP01) fails, and is fenced out of the sysplex (our job's not having a good day, is it?)

   - The job is restarted again, on BP02. It re-registers again.

2) The display results now show two restarts, and reflect the change of systems associated with this element.

ARM restart, cross system

Batch job still running after restart, on BP01

```
$HASP373 BEPEJOB1 STARTED - INIT A - CLASS F - SYS BP01
IEF493I  BEPEJOB1 STARTED - TIME=00.53.14
```

(job registers with ARM)

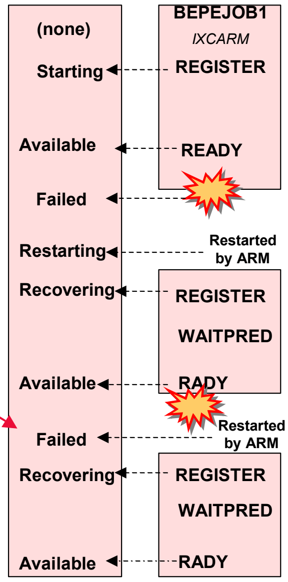
BP01 fails, is fenced out of sysplex, on BP02

```
IXC812I JOBNAME BEPEJOB1, ELEMENT BEPEJOB1 FAILED DUE TO
        THE FAILURE OF SYSTEM BP01
        THE ELEMENT WAS RESTARTED WITH PERSISTENT JCL

$HASP373 BEPEJOB1 STARTED - INIT A - CLASS F - SYS BP02
IEF493I  BEPEJOB1 STARTED - TIME=01.05.23
```

(job re-registers with ARM)

D XCF,ARMSTATUS,DETAIL

```
IXC392I 01.08.32 DISPLAY XCF
   ARM RESTARTS ARE ENABLED
   --------------- ELEMENT STATE SUMMARY ---------  TOTAL- -MAX-
   STARTING AVAILABLE FAILED RESTARTING RECOVERING
          0         1      0          0          0      1    20
   RESTART GROUP:DEFAULT        PACING  : 0       FREE CSA:   0    0
   ELEMENT NAME:BEPEJOB1        JOBNAME :BEPEJOB1 STATE   :AVAILABLE
     CURR SYS :BP02             JOBTYPE :JOB      ASID    :015E
     INIT SYS :BP01             JESGROUP:BPPLEX01 TERMTYPE:ALLTERM
     EVENTEXIT:GOSSIP99         ELEMENTYPE:*NONE* LEVEL   :   2
     TOTAL RESTARTS :      2      INITIAL START:05/30/09 00.53.14
     RESTART THRESH : 0 OF 3      FIRST RESTART:05/30/09 00.56.54
     RESTART TIMEOUT:    300      LAST RESTART :05/30/09 01.05.23
```



MPOR - 05 - 58

1.52 ARM considerations

Things to consider when using ARM:

- It's quite easy to set up an ARM environment if you just take the defaults, but the defaults are potentially dangerous. There is no free CSA requirement, for example, which won't help if you try a cross-system restart of fifteen CICS regions. Also, having everything in a single restart group with no RESTART_PACING could overwhelm a system. You should code a policy to cover your specific requirements if you want to use ARM.

- If you plan to restart subsystems like DB2 on other systems, you'll need to ensure all subsystems are defined to all systems

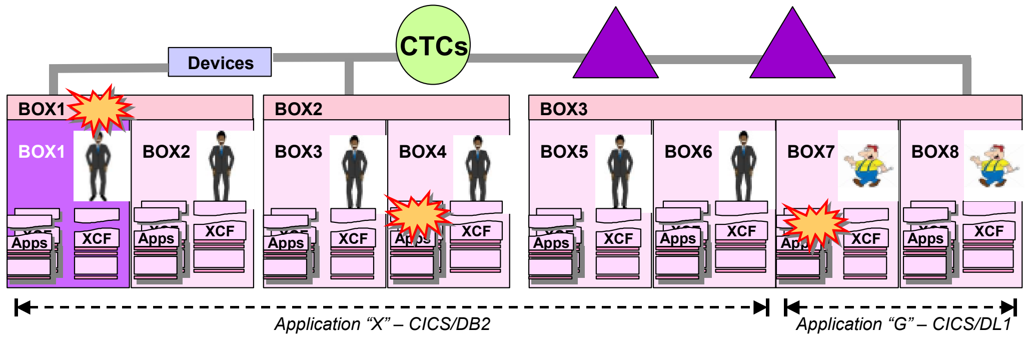
- In your policy, code a default RESTART_GROUP with RESTART_ATTEMPTS(0). If you don't do so, then all registered elements not listed explicitly in you policy will be restartable (which may not be what you had planned)

- Consider how you should deal with transaction based work like CICS. If you lost BP01, for example, you might lose ten of the 60 CICS AORs servicing your CICS/DB2 application. Should you restart those AORs across the other systems or will the remaining 50 AORs be able to cope with all the transactions?

- Again, if you lose BP01, you will lose a DB2 with its IRLM. There is no point in restarting that DB2 on one of the other systems already running another DB2 in the same group, but you'll still need to restart it so the IRLM can recover any retained locks created when the system fell over. Once this is done you can shut that DB2 down again.

Setting up a proper ARM policy takes a lot of planning!

ARM considerations



ARM considerations

- Don't run with the defaults, create an installation policy
- Include a default RESTART_GROUP definition with RESTART_Attempts(0) to exclude restarts for non-explicit elements
- To use cross-system restart, pre-define all subsystems to all systems
- If you lose BP01 above, can you support all your CICS/DB2 transactions on the remaining regions, or should you restart the lost regions across the other images?
- Subsystems like IRLM may need to be cross-system restarted to recover lost resources, but are then no longer required

A proper ARM environment requires a lot of planning!

MPOR - 05 - 59

1.53 Summary

The first rule about sysplex recovery processing is simple - **duplicate everything in sight!**

- sysplex timers
- CTC connections
- CF channels
- Coupling Facilities
- Couple Data Sets

These should all (ideally) have alternates available and /or configured. In most cases if an alternate is available, no action is required if the original is lost and recovery is automatic.

System isolation

Remember, if you can't communicate in the sysplex, you can't continue. If you **don't** have duplicate components, systems will be removed from the sysplex via the 0A2 wait state in the event of connectivity failures and status update missing conditions.

Sysplex Failure Manager

The z/OS **Sysplex Failure Manager (SFM**) can, via its SFM Policy definitions:

- **automatically** reconfigure the sysplex and isolate a failing system if these connectivity failure or status update missing conditions occur

- automatically **rebuild failed structures** in the above circumstances using the *rebuild percentage* and *system weights* values as its guidelines

Automatic Restart Manager

The **Automatic Restart Manager (ARM)** can be used to restart failed batch jobs or started tasks based on criteria in the ARM policy:

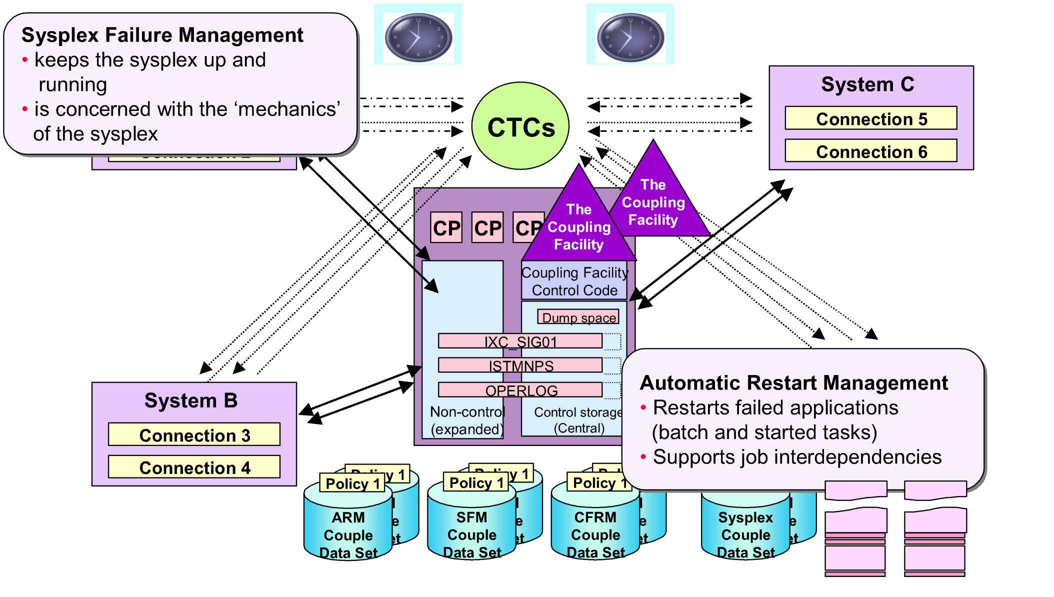
- restart can be performed when the job or the system fails, or both

- **restart levels** control the order in which jobs are restarted

- **restart group** definitions specify actual restart requirements for individual jobs

- jobs/tasks must issue **IXCARM** to **register** with ARM

Summary



MPOR - 05 - 60

1.54 Exercise 5

PART 1

1. Deactivate a CF. The instructor will tell you which one to choose.

2. What has happened to the following structures?

   IXCSIG1
   Has signalling continued?

   IRRXCF00_P001
   Is RACF still in Datasharing Mode?

   JES2CKPT
   Has the backup structure been used?

   ISTGENERIC
   Has duplexing stopped?

3. Activate the CF.

4. Re-allocate the structures. Is everything back to normal?
   (**Hint**: Check the original Checkpoint structure. Ensure that it is de-allocated.)

PART 2 - SFM Active

5. Change the Interval time to 20 and the Opnotify to 23 on RSMB.

6. Start the SFMPOLICY, SFMPOL1.

7. Check that it has started.

8. Check the COUPLE parameters on RSMB.

9. Perform a System Reset on RSMB on the HMC.

10. What is the current status?

11. Ensure RSMB is removed from the sysplex.

12. Did SFM partition RSMB?

13. What ‘Wait State’ did RSMB enter?

14. Re-IPL RSMB with the correct LOADPARM.

PART 3 - SFM Not Active

15. Stop the SFM policy.

16. Perform the same steps as Part 2.

17. What is the difference between Part 2 and Part 3?

PART 4

18. Move all the signalling structures to RSMICFB.

19. De-activate the RSMICFB.

20. What has happened?

21. Is everything OK?

22. Activate RSMICFB.

23. Move one of the signalling structures back to the other CF.

Student notes

Student notes