

Resilience

Keeping Batch Going, Whatever

Anna Dawson
IBM (UK) Ltd

November 2019
Session **NF**

Place your
custom session
QR code here.
Please remove
the border and
text beforehand.



Resilience

- the capacity to recover quickly from difficulties; toughness.
- the ability of a substance or object to spring back into shape; elasticity



Topics

- Auto-Recovery – Getting a job running again without delay
- Alternate Workstation – Moving workload to another LPAR
- Current Plan backup and recovery – Current Plan Corruption
- Hot Standby – Controller LPAR failure
- Remote Standby – Controller Site failure

Auto-Recovery

Getting jobs running again without delay

Auto-Recovery – Overview

When a job within an application fails, automatic recovery can be invoked to redress the situation.

Automatic Recovery statements in a jobs' JCL can cause the job to be restarted, rerun or altered in some way, before rerunning automatically, when the job fails matching an error coded on the statement.

The actions will only take place if the failure exactly matches one of the explicitly coded failure criteria.

The Automatic Recovery task can be switched on and off via the services panel. Apart from that interaction with the IzWS dialogs the control for automatic recovery is contained within a jobs JCL.

IBM Z Workload Scheduler retrieves the JCL for automatic recovery from the JCL repository (JS) data set. This means that automatic recovery can take place only for jobs or started tasks submitted by IBM Workload Scheduler for z/OS.

The automatic recovery function takes over when a job or started task ends in error. At that time, the following information is available:

- ❑ The error code for the operation. This can be:
 - The abend code of an abending step
 - The return code of the last step
 - An error code set by IBM Workload Scheduler for z/OS, such as JCLI, CCUN, JCL, CLNO, CLNA, CLNC, CAN, PCAN, CLNP, OFxx, or OSxx
 - An error code set by the job completion checker
- ❑ The name of the abending step, if the error is associated with a step.
- ❑ Step completion codes and step names for all steps executed. The step completion code is either an abend code or a return code.

Note: Automatic recovery is not applicable for error codes, such as OSUP, that refer to jobs that have not reached the job queue.

Auto-Recovery – Actions

The JCL can contain multiple recovery actions to cater for different failures and their recovery actions.

The actions that can be taken are:

- Restart at failed operation (with or without JCL changes)
- Restart at an earlier CPU operation in the occurrence
- Add another application into the CP, as a predecessor to the failed job/application or not
- Release an external successor dependency
- Restart at the failed or another step (with or without JCL changes)
- Leave in error

The JCL can be amended by deleting steps, adding procedures, calling an exit to read/amend the job and by selecting the restart point.

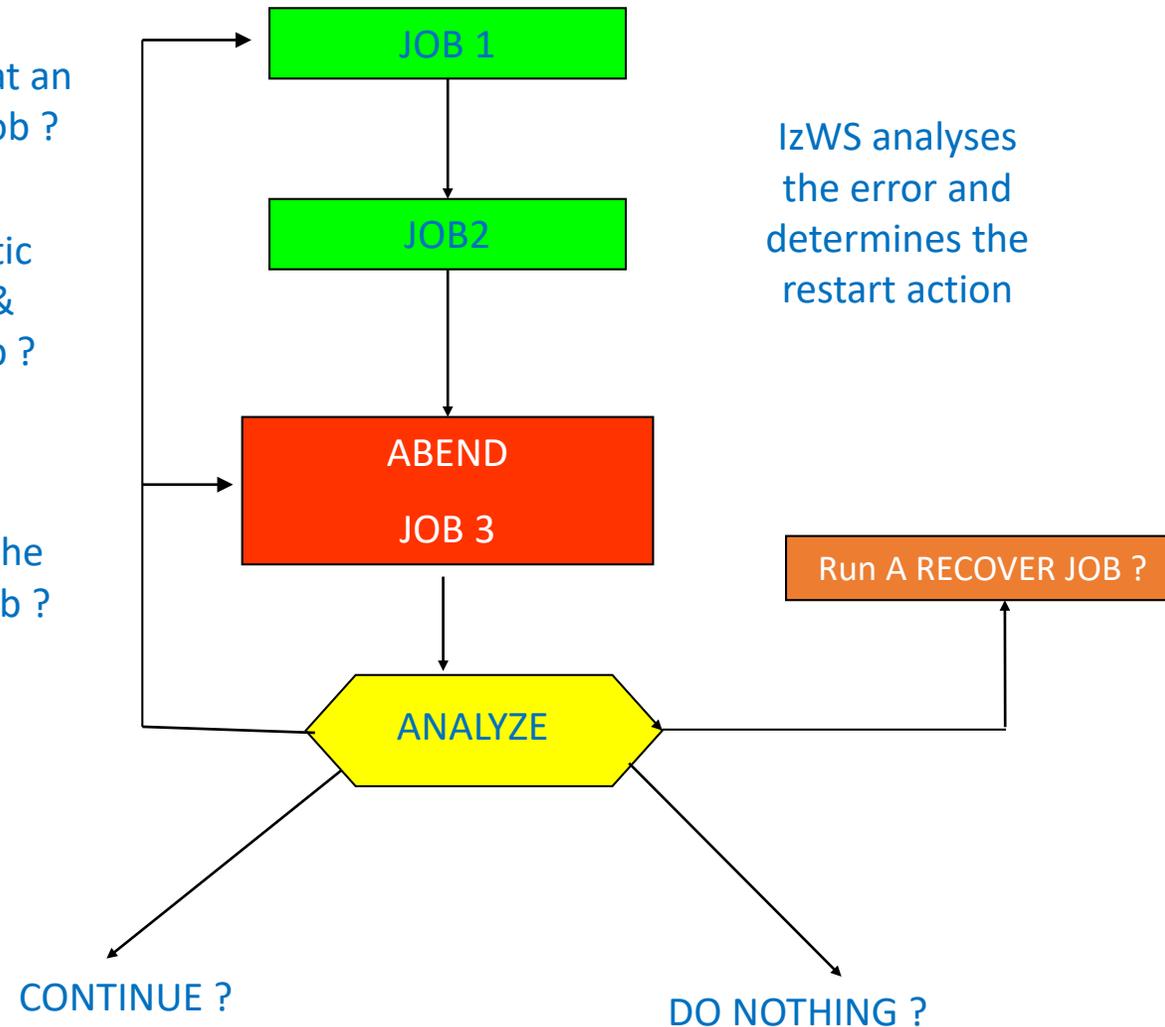
the recovery instructions must be inserted between the JOB statement and the first execution step

Restart at an earlier job ?

Automatic Restart & Clean-up ?

Restart the failing job ?

IzWS analyses the error and determines the restart action



Auto-Recovery – Syntax

Syntax to identify failure match

```

/**%OPC RECOVER
  <_,'_____ |_)
  _ERRSTEP=(  _stepname_____ |_)
              |_stepname.procstepname_|
              |_stepname-stepname_____ |
  <_,'_____ |_)
  _JOBSTEP=(  _abendcode_____ |_)
              |_casecode_____ |
              |_returncode_____ |
              |_returncode1-returncode2_|
  <_,'_____ |_)
  _STEPSTEP=(  _abendcode_____ |_)
              |_casecode_____ |
              |_returncode_____ |
              |_returncode1-returncode2_|
  _TIME=(  _hhmm-hhmm_____ )

```

Note: Automatic recovery is very flexible, however it is advisable to test different situations – IzWS supplies a test program in SEQQSAMP(EQQRETWT) called EQQRETWM that does a wait and also simulates different failure codes

Syntax to determine recovery actions

```

  _DELSTEP=(  _stepname_____ |_)
              |_stepname-stepname_|
  <_,'_____ |_)
  _ADDPROC=(  _procedure name_____ |_)
  <_,'_____ |_)
  _RESSTEP=(  _stepname_____ |_)
              |_stepname.procstepname_|
              |_*_____ |
              |_%_____ |
              |_checkid_____ |
  _CALLEXIT=(  _program name_____ )
  _RESTART=(  _Y_____ |_)
              |_YES_____ |_)
              |_N_____ |_)
              |_NO_____ |_)
  _RESJOB=(  _job name_____ )
  <_,'_____ |_)
  _ADDAPPL=(  _application name_____ |_)
  <_,'_____ |_)
  _RELSUCC=(  _application name_____ |_)
  _ALTWS=(  _workstation name_____ )
  _ALTJOB=(  _job name_____ )

```

Auto-Recovery – Example

- Auto-recovery actions are specified by coding recover directives in the JCL

//*%OPC RECOVER

Depending on the combinations, syntax may be complex and should be tested

Refer to The Managing the Workload manual and the chapter on Automatic recovery of jobs and started tasks

- Example code:

```
//T184727A JOB .....
//*%OPC RECOVER JOBCODE=S*37,ADDPROC=REORG
//*%OPC RECOVER JOBCODE=(*,16-4095),ADDPROC=REORG,
//*%OPC RESTART=NO
//*%OPC RECOVER JOBCODE=12
//STEP01 EXEC PGM=ATTACHBA,REGION=4K
//TESTIN DD .....
//.....
```

- IBM Z Workload Scheduler begins the automatic recovery process by
- scanning the job for the first `//*%OPC RECOVER` statement where:
 - The step name matches the name of the failing step from the operating system.
 - The error code matches the error code from the job and started-task tracking function.
 - The return code matches the step return codes or abend codes from the job and started-task tracking function.
 - The RECOVER statement is unconditional (it specifies no step name, error code, return code, or abend code).
- This means you should place the RECOVER statements with the most restrictive matching conditions before the RECOVER statements that deal with more general cases.

Alternate Workstation

Getting jobs running again without delay

Workstation Status

To run work, a computer workstation must be active and open. A workstation is

- **active** if the controller can communicate with it. An active computer workstation
- can be **open or closed**. A closed computer workstation is not eligible to have work scheduled on it even if the controller can communicate with the workstation.

Workstation status can be changed dynamically (either manually, using IBM Z Workload Scheduler panels, or automatically, in response to changes in your systems).

An inactive workstation can have one of the following statuses:

- **FAILED** - The operating system has detected a failure on the system that the workstation is defined on.
- **OFFLINE** - Communication is lost between the controller and tracker on the system that the workstation represents. This might be because the tracker is not yet started or because it ended abnormally.
- **UNKNOWN** - the scheduler has detected that the workstation is inactive, but no other diagnostic information is available.

Redirecting work to alternate workstations

The scheduler supports the redirecting of work from one workstation to another.

If a workstation becomes inactive, you can specify, for each of its open intervals, an alternate workstation where work will be redirected.

At any time, you can manually redirect the work to the workstation by using the MODIFYING THE CURRENT PLAN panel (option 5.5 from the main menu).

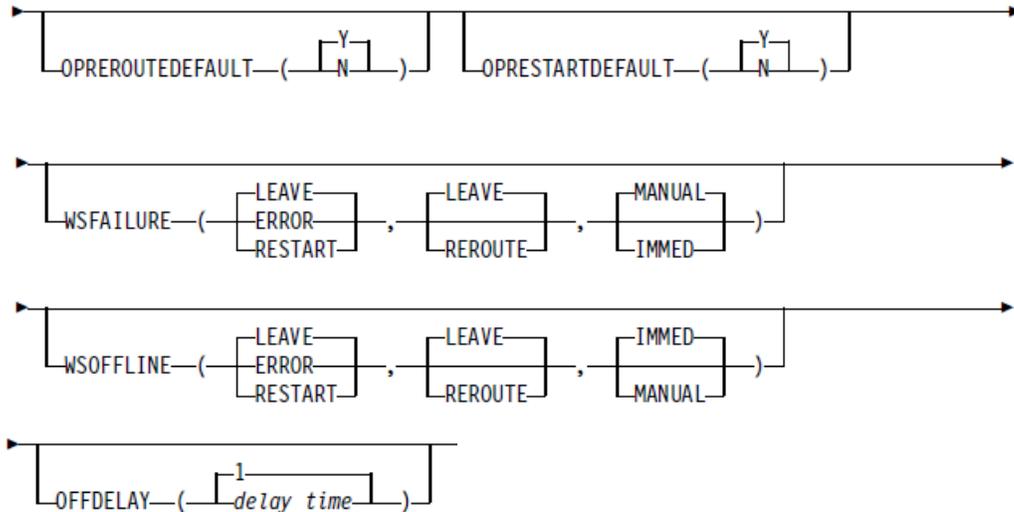
Note: This function does not apply to either fault-tolerant or virtual workstations.

What happens when the workstation fails or goes offline depends on the values you provided to the JTOPS parameters WSOFFLINE and WSFAILURE

It also depends on the operations values for RESTART and REROUTE

There are JTOPTS parameters that set a default value that can be overridden in the job definition

JTOPTS



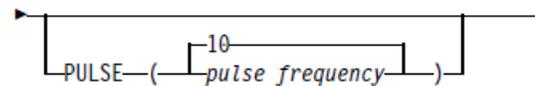
Default value for the restart and reroute options in the job option panel

Actions to take when the Controller recognises that the workstation destination has failed

Actions to take when the Workstation becomes offline – that is when the tracker and controller lose connection

How long should the controller delay actions after setting a workstation offline (includes waiting for tracker connection)

ROUTOPTS

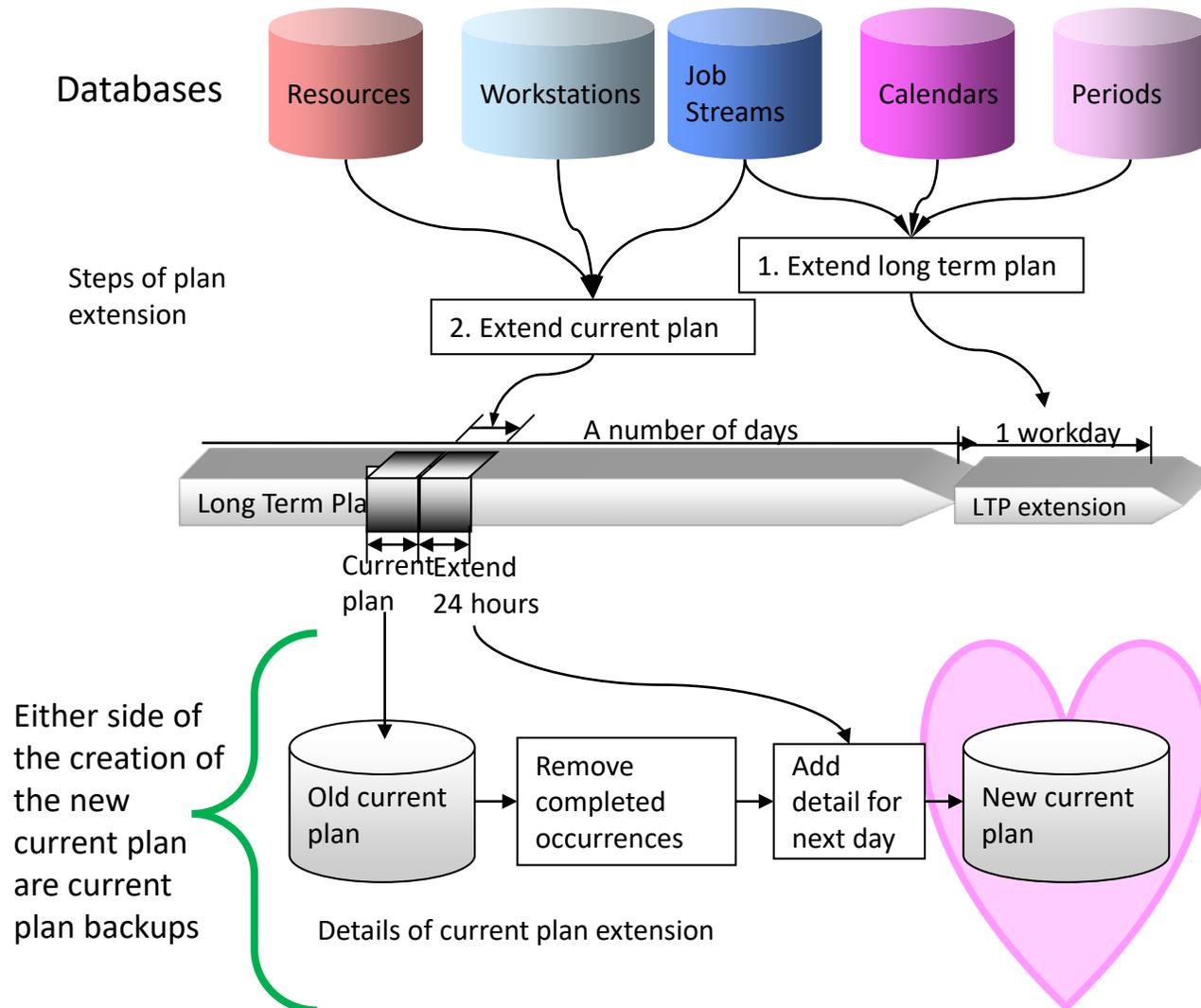


Time between pulses (Controller / Tracker) 2 consecutive missed pulses will trigger an “offline” event

Current Plan Backup & Recovery

Ensuring swift recovery

Long Term and Current Plan Extend Process

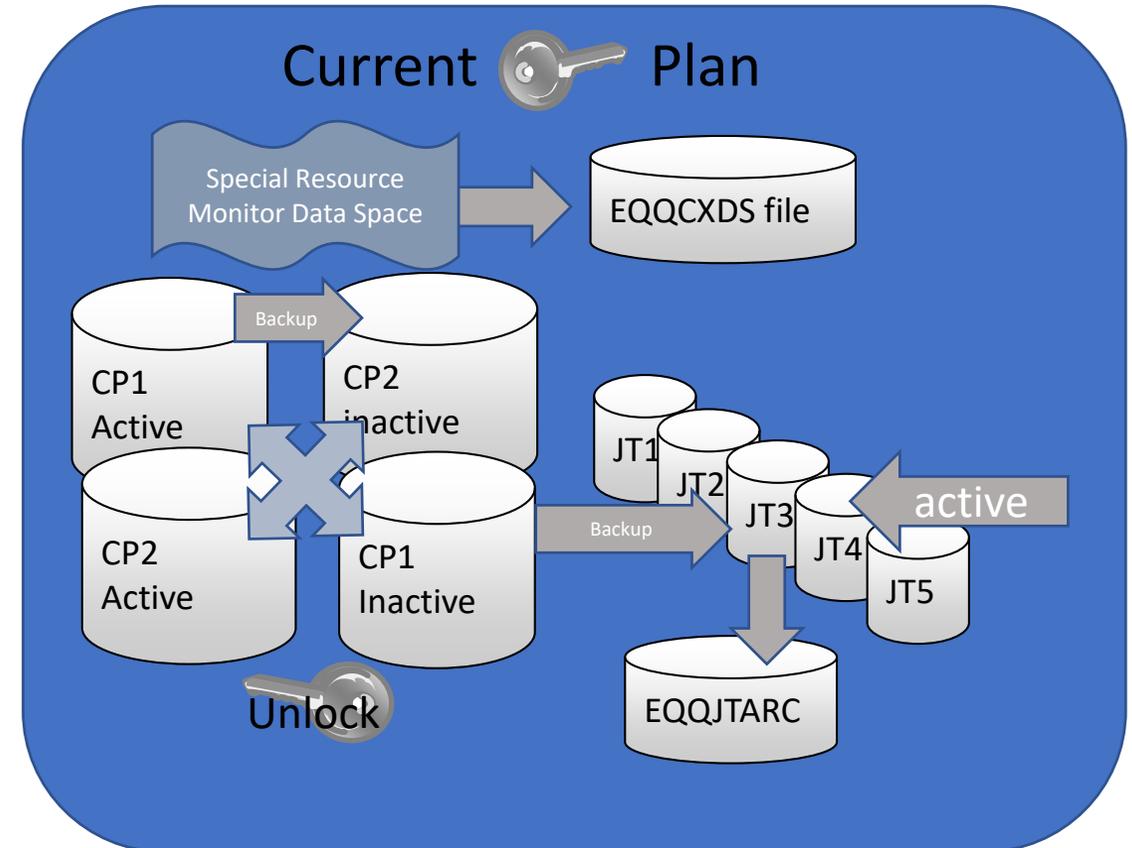


A current plan backup is performed when:

- During normal processing, according to the value of the BACKUP parameter of the JTOPTS initialization statement. This parameter specifies the number of current plan updates that must occur before a current plan backup is performed (or it says NO).
- When the BACKUP command is issued specifying the current plan resource.
- Immediately before normal termination of the Controller
- When the Controller detects that a daily planning batch job has started.
- After a daily planning batch job has created a new current plan.
- After current plan recovery processing has successfully re-created an up-to-date current plan.

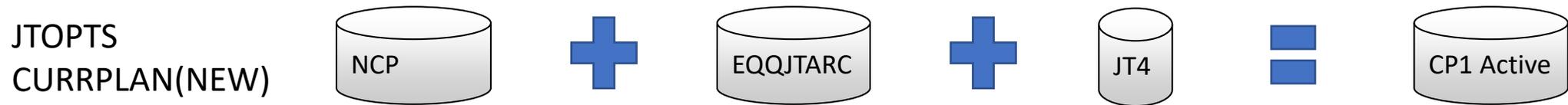
Step-by-step current-plan backup process

1. The Controller locks the current plan to prevent it from being updated during the backup process. During the backup, events are queued in storage. Those working with the current plan may experience a short delay.
2. The CX data space (EQQCXDS – Special Resource Monitor) is backed up to DASD.
3. The backup current plan is erased.
4. The active current plan is copied to the backup current plan. The contents of the two are now identical.
5. The data sets are switched. The backup current plan becomes the active and the active becomes the backup.
6. A current plan backup record is written to the job-tracking log, and the next available job-tracking log becomes active. The corresponding dual job-tracking log is also switched.
7. The current plan is unlocked. Normal processing continues. Events queued in storage start to update the active current plan. Panel users' requests are processed.
8. The data from the now inactive job-tracking log is appended to the job-tracking archive log. The inactive job-tracking log is emptied for future use.
9. If you activated the extended auditing feature, the data from the now inactive extended-auditing data log is appended to the extended-auditing archive log. The inactive extended-auditing log is emptied for future use.



Why does this Current Plan Backup = Resilience?

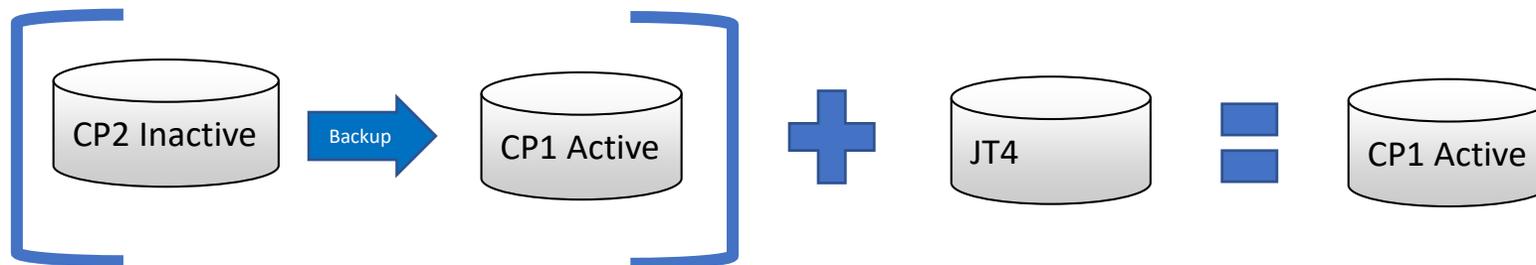
- The NEW Current Plan creates a base point for the current plan.
- As part of the current plan extend the JTARC file is emptied into the EQQTROUT file and the JT1 – JT_x files start collecting the Job Tracking events and updating the EQQTARC whenever the CP Backup processes
- From the point the NCP (New Current Plan) is created, it plus the EQQTARC and the currently recording JT_x file, provide a forward recovery log to rebuild the current plan to point of failure.



- Normally you start the Controller with the JTOPTS parameter CURRPLAN(CURRENT) and the Controller starts up using the Current Plan as it was when the Controller stopped
- When you start the Controller with the JTOPTS parameter CURRPLAN(NEW) the Controller starts using the NCP (New Current Plan) and applies all the events found in the EQQTARC, and if there was no clean shutdown, the events in the JT_x tracking log that was active.
- This allows the Current Plan to be rebuilt to the point where the Controller was lost (assumption is some sort of failure).
- A suggestion would be to also start with submission turned off – JOBSUBMIT(NO)
- Once the Current Plan is rebuilt the Controller will re-establish communication with the trackers and resynchronize with them, receiving and processing all the events since it was down.
- Then batch can continue

Cancel, Cancel, Cancel

- If you suspect the Current Plan has been corrupted, do not stop the Controller. Stopping the Controller will cause it to synchronise the CP1 and CP2 files and archive the JT_x file, propagating the corruption to the inactive CP_x
- The correct action is to **CANCEL** the Controller
- This prevents the synchronisation
- When the Controller is restarted with the JTOPTS parameter CURRPLAN(CURRENT) it will “know” that the CP1 and CP2 are not in synch and will perform a recovery start up which is quicker than restarting from the NCP



Hot Standby

Coping with Controller or LPAR failure

Controller Modify Command */F procname,modifyoption*

To enable Hot Standby the Controller data should reside on DASD available across the LPARS where the primary and secondary Controllers are located.

There can be several standby Controllers within a SYSPLEX

If the primary Controller will be unavailable due to LPAR unavailability the Controller can be switched to the alternate using the TAKEOVER command before the LPAR is brought down

If the LPAR is lost unexpectedly – the first alternate Controller in the SYSPLEX that notices the loss will ask to takeover.

/F OPCA,TAKEOVER

Orders a standby controller to take over the functions of the controller.

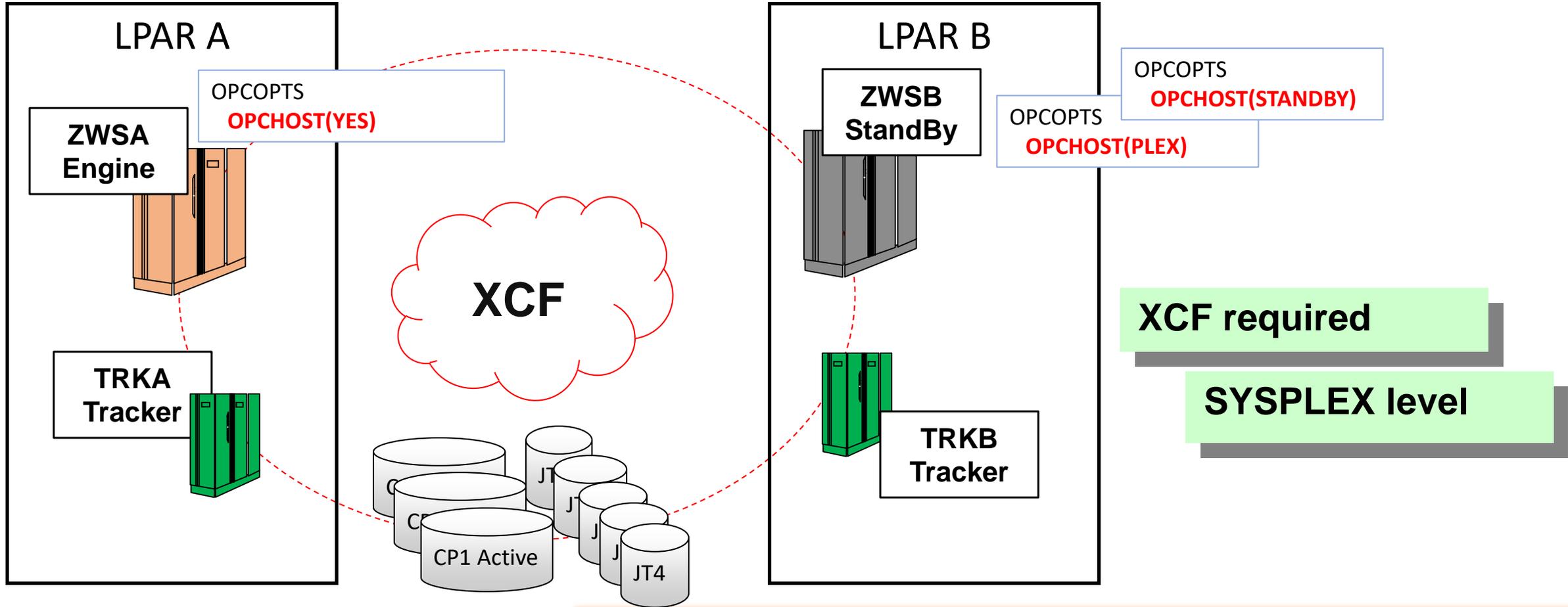
This command is valid only when both systems are part of the same XCF group, and no controller is active. You can use this command only for IBM Z Workload Scheduler address spaces where OPCHOST(STANDBY) is specified on the OPCOPTS initialization statement.

Note: Takeover can occur automatically if you have specified the TAKEOVER keyword on the XCFOPTS initialization statement of a standby system.

Note: Because Servers must run on the same system as the Controller, ensure all the servers required are started with the Controller using the SERVERS parameter

HOT Standby - Configuration

High Availability within a SYSPLEX



TAKEOVER automatic (SYSTEM / Controller failure)

Remote Hot Standby

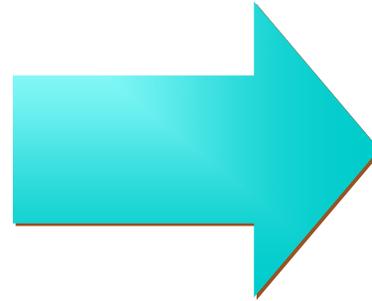
Coping in a disaster

Challenge for Remote HOT Standby

Enable **sysplex** failover
Planned and **unplanned** failures
Shortest RTO and RPO

Mirror plans and data to remote site in near real time

JT, CP1/2 NCP CX NCX XD1/2
 AD WS RD OI

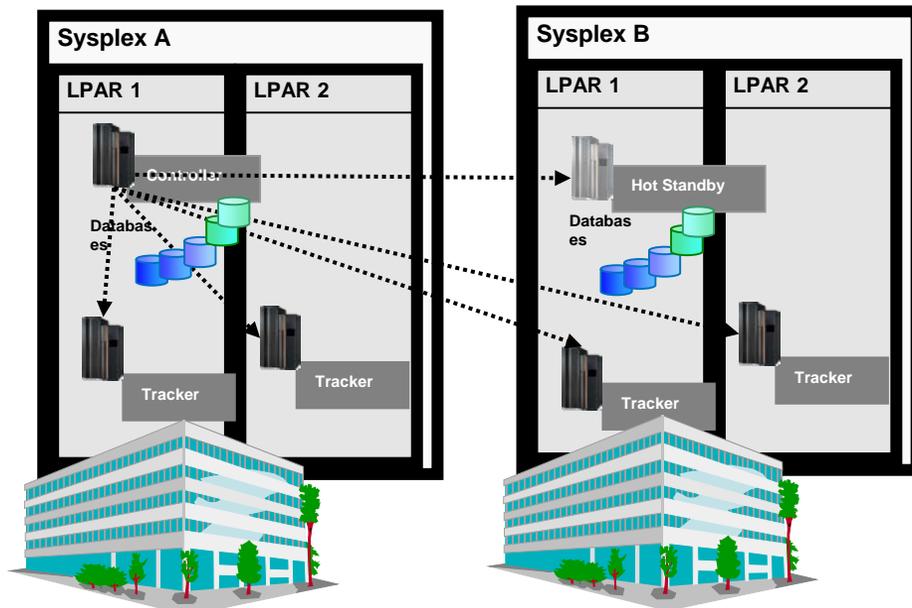


A backup Controller

Planned and unplanned failover

- Switch Controller
- Switch Controller and trackers

Reverse switch from backup to primary



Pre-requisites

TCP/IP connection
 between sites

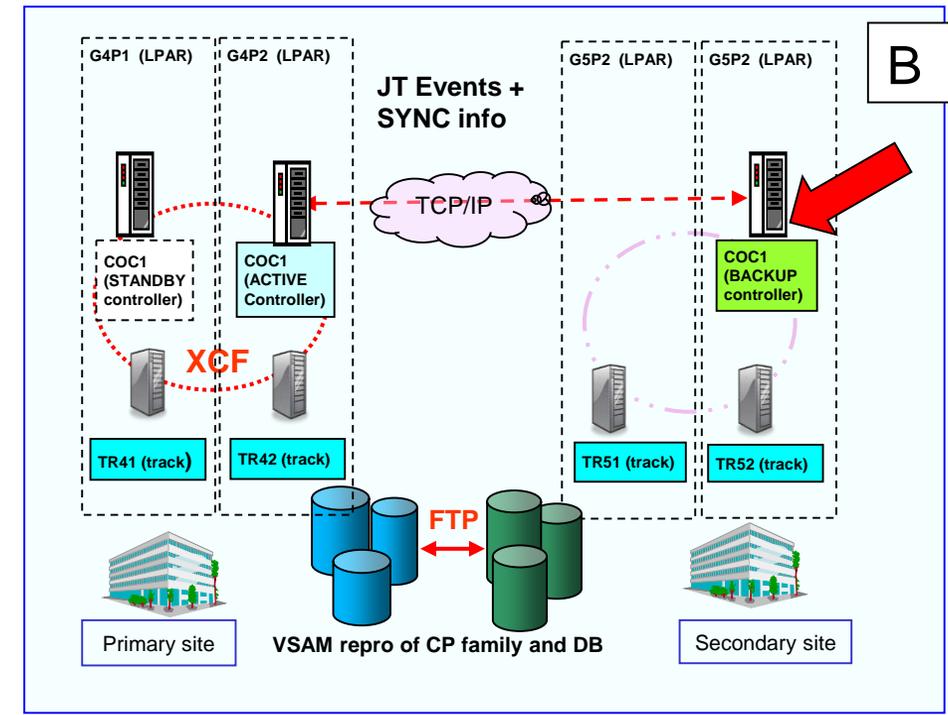
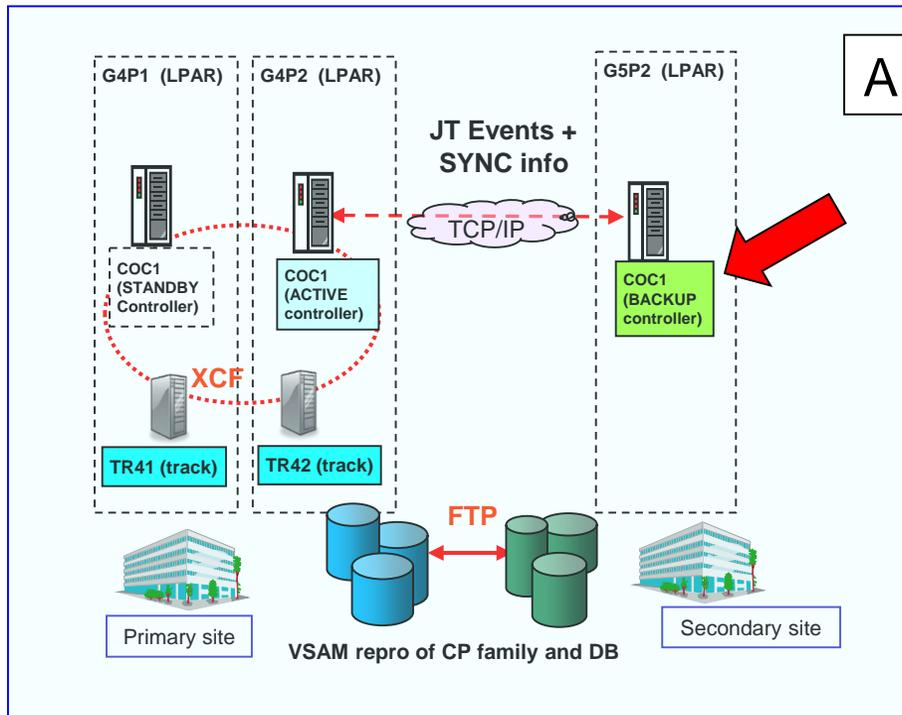


Solution – remote BACKUP Controller



The **backup Controller** component is a TWSz Controller installed on a remote sysplex that runs in backup operational mode. It is connected through TCP/IP to the primary Controller and receives plan files and event updates for plans and databases from the primary Controller.

The failover scenario supported includes the switch of the Controller to the remote backup (Fig. A) or the switch of both Controller and Trackers on a remote site for a whole sysplex recovery. (Fig. B).



Configuration considerations:

- TCP/IP communication between sites:

- primary and backup Controller
- Controllers and remote Trackers
 - z/OS Tracker connection type can be only XCF or TCP/IP



- FTP to transmit data set between remote sites

The backup Controller extends the Hot Standby feature outside of one SYSPLEX.

Both Controllers must have the same configuration, the same subsystem name and the same VSAM data sets name.

A configuration with a backup Controller supports:

- z/OS Trackers
- IzWS agents (aka zcentric agents)
- Dynamic Domain managers
- Cross Dependencies

In JES3 environment Controller should run on GLOBAL.

- If running on LOCAL workaround via ETT definition is needed to track conversion phase JCL errors in repro/restore jobs.

A product solution for IzWS failover without need of HW mirroring (only TCP/IP communication and FTP is needed).

Short time needed to switch:

- A modify command to turn a backup Controller into a primary one.
- z/OS and zCentric agents reconnect automatically after the switch.

Customisation



Customize

To use the new function you need to:

- ✓ Run EQQJOBS
- ✓ Allocate needed data sets
(backup Checkpoint, GDG roots, backup controller data sets, new ST VSAM)
- ✓ Customize the Transfer and Restore procedures (including FTP usage)
- ✓ Copy the Transfer and Restore procedures into the Controllers JOBLIB
- ✓ Set up Controllers and Trackers initial parameters according to your specific configuration
- ✓ Set up Controllers and Trackers start JCL
- ✓ Transfer to remote site (using the provided samples) the Data Base (EQQSENDDB), JS files (EQQSJS1/EQQSJS2) and JOBLIB.
- ✓ Restore the just transferred data set on remote site
- ✓ **At this point you can start for the first time the backup Controller!**

Please submit your session feedback!

- Do it online at <http://conferences.gse.org.uk/2019/feedback/nf>
- This session is **nf**

1. What is your conference registration number?

 This is the three digit number on the bottom of your delegate badge

2. Was the length of this presentation correct?

 1 to 4 = "Too Short" 5 = "OK" 6-9 = "Too Long"

1 2 3 4 5 6 7 8 9

3. Did this presentation meet your requirements?

 1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

4. Was the session content what you expected?

 1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

Place your custom session QR code here. Please remove the border and text beforehand.

Backup Slides

Alternate Workstation

WSFAILURE(ERROR|RESTART|LEAVE, REROUTE|LEAVE, IMMED|MANUAL)

Defines the actions to be taken when a workstation failure occurs. A workstation is set to failed status either when there is no communication with the z/OS system or if it is set manually in the IBM Z Workload Scheduler dialogs. Workstations that specify a user-defined destination ID can have failed status reported by the WSSTAT command, or EQQUSIN or EQQUSINW subroutine.

The WSFAILURE parameter contains three keyword values:

The first keyword value defines the restart actions to be taken when a workstation fails and the started operation is restartable.

Specify ERROR to set started operations to ended-in-error status. The error code will be OSSI, OSSQ, OSSS, or OSSC. Operations whose error code is OSSS have a step code of OSYS.

Specify RESTART to immediately reset the started operations at this workstation to ready. Specify LEAVE to leave started operations on a failed workstation in started status; this is the default value.

Note:

1. If you set the restartable option of an operation to NO, the operation is not processed. It remains in the started status.
2. Consider this note if you use a standby controller, running with the TAKEOVER(HOSTFAIL) parameter in the XCFOPTS statement. If you selected the WSFAILURE(RESTART,REROUTE,IMMED) options and the controller abnormally ends or is cancelled, while the LPAR that the controller is running on remains active, jobs might be submitted again even though they are currently running, resulting in the same job being run twice.
3. For dynamic and remote engine workstation types, this keyword supports only the value LEAVE. If you specify any other value, it is forced to LEAVE.

The second keyword value defines reroute actions for workstation failure situations.

Specify REROUTE to route operations, whose reroutable option is YES, to the alternate workstation.

Specify LEAVE to leave operations to be scheduled on the original workstation; this is the default value. Rerouting does not occur for these operations.

The third keyword value defines the action to be taken when a workstation becomes active again after a failure situation.

Specify IMMED to automatically set the status of the workstation to available and withdraw any rerouting *immediately* when an event indicates that the workstation is operational.

Specify MANUAL to indicate that the status of the workstation should be changed manually when a *workstation available* indication is received; this is the default value. IBM Z Workload Scheduler issues an MLOG message to inform the operator that the event has been received.

WSOFFLINE(ERROR|RESTART|LEAVE, REROUTE|LEAVE, MANUAL|IMMED)

Defines the actions that are to be taken when a workstation offline situation occurs (unless the workstation is 'waiting for connection' at start and no previous offline situation occurred). This means that the controller cannot communicate with the tracker at the destination defined for the workstation. This might occur because the tracker has not been started yet (having experienced a previous offline condition status) or has ended abnormally, or because the controller has not received an ID event from the destination for two consecutive pulse intervals. Pulse intervals are specified by the PULSE keyword of ROUTOPTS.

Workstations that specify a user-defined destination ID are set to offline status when IBM Z Workload Scheduler is started. Offline status for these workstations can also be reported by the WSSTAT command or the EQQUSIN or EQQUSINW subroutine.

The WSOFFLINE parameter contains three keyword values:

The first keyword value defines restart actions for a workstation whose status has been changed to offline. Specify ERROR to set started operations, whose restartable option is YES, to ended-in-error status. The error code will be OFSI, OFSQ, OFSS, or OFSC. Operations whose error code is OFSS, have a step code of OFFL. Specify RESTART to immediately reset the started operations at this workstation to ready. Specify LEAVE to leave started operations at an offline workstation in started status; this is the default value.

Note:

1. If you set the restartable option of an operation to NO, the operation is not processed. It remains in the started status.
2. Consider this note if you use a standby controller, running with the TAKEOVER(HOSTFAIL) parameter in the XCFOPTS statement. If you selected the WSOFFLINE(RESTART,REROUTE,IMMED) options and the controller abnormally ends or is cancelled, while the LPAR that the controller is running on remains active, jobs might be submitted again even though they are currently running, resulting in the same job being run twice.
3. For dynamic and remote engine workstation types, this keyword supports only the value LEAVE. If you specify any other value, it is forced to LEAVE.

The second keyword value defines reroute actions for workstation offline situations.

Specify REROUTE to route operations, whose reroutable option is YES, to the alternate workstation.

Specify LEAVE to leave operations to be scheduled on the original workstation; this is the default value. Rerouting does not occur for these operations.

The third keyword value defines the action to be taken when a workstation becomes active again.

Specify MANUAL to indicate that the status of the workstation should be changed manually when a *workstation available* indication is received. IBM Z Workload Scheduler issues an MLOG message to inform the operator that the event has been received.

Specify IMMED to automatically set the status of the workstation to available and withdraw any rerouting *immediately* when an event indicates that the workstation is operational; this is the default value.

Backup Slides
Remote Hot Standby



EQQJOBS (IzWS Installation aid)

```
EQQJOBSD ----- Create sample job JCL -----
Command ==>

BKPT configuration:  Y      (Y= Yes ,N= No)
Backup data set name ==> TWSSSD.CWSA.BKPT_____

BKPTOPTS statement procedure names:

DUMP Procedures for:
NCP,NCX,NXD Files   ==> ROZSENCP
CX,CP1,XD1 Files    ==> ROZSECP1
CX,CP2,XD2 Files    ==> ROZSECP2
LTP File             ==> ROZSENLT

RESTORE Procedures:
NCP,NCX,NXD Files   ==> ROZRENCP
CX,CP1,XD1 Files    ==> ROZRECP1
CX,CP2,XD2 Files    ==> ROZRECP2
LTP File             ==> ROZRENLT

Press ENTER to create sample job JCL
```

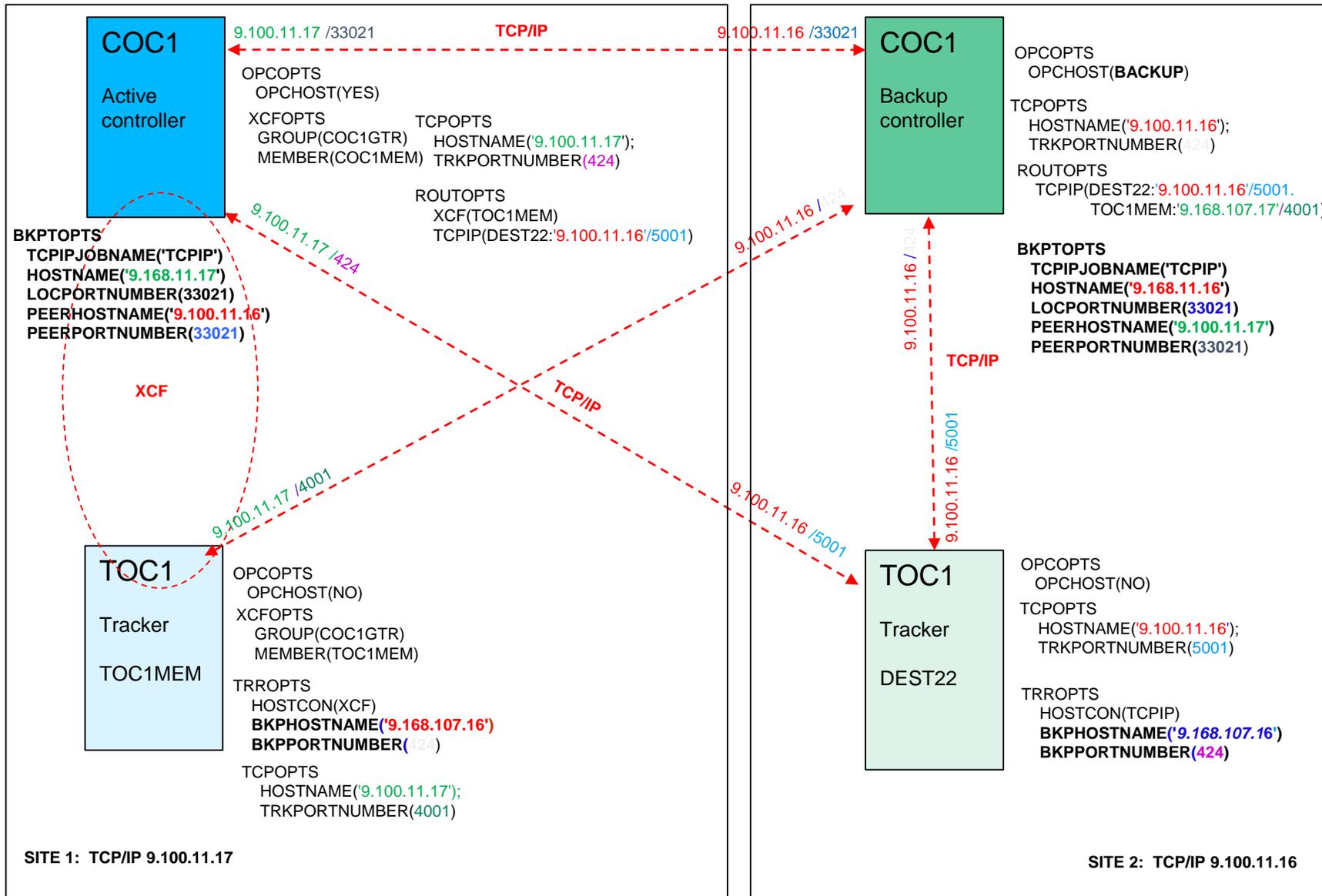
Input

```
BKPTOPTS TCPIPJOBNAME('TCPIP')
          HOSTNAME(lochname)
          LOCPORTNUMBER(lochport)
          PEERHOSTNAME(peerhname)
          PEERPORTNUMBER(peerhport)
          NCPRESTPROC(ROZRENCP)
          CP1RESTPROC(ROZRECP1)
          CP2RESTPROC(ROZRECP2)
          LTPRESTPROC(ROZRENLT)
          NCPDUMPPROC(ROZSENCP)
          CP1DUMPPROC(ROZSECP1)
          CP2DUMPPROC(ROZSECP2)
          LTPDUMPPROC(ROZSENLT)
```

```
OPCOPTS  OPCHOST(BACKUP)
          APPCTASK(NO)
          ERDRTASK(0)
```

Output

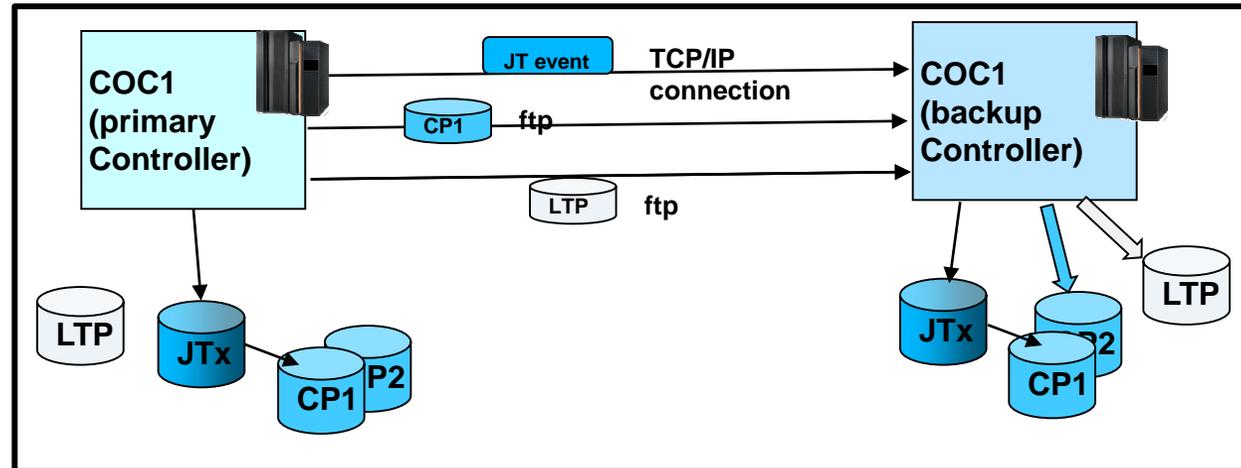
Customization example



How the backup Controller is up to date with the primary:

- receives **JT events** that are applied to its DB/plans
→ normal processing

JT event



- receives and restores the **plan files (CP and LTP)**
→ re-synchronization processing due to
 - start up
 - new plans created on primary Controller
 - ** DP batch run
 - ** LTP batch run
 - event lost impossible to be resent (not in current JT)



Synchronization - backup Controller start up

connect

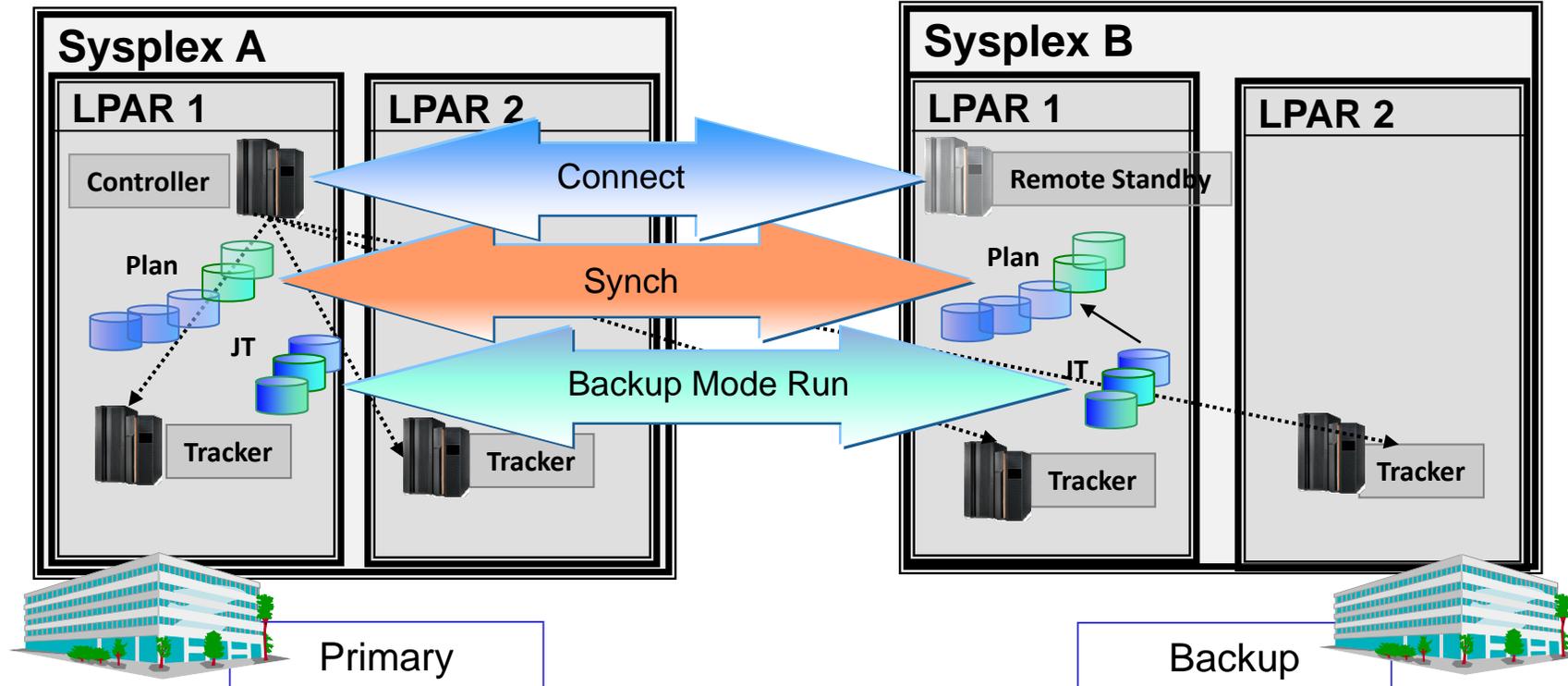
Connects to active

synchronize

*Synchronize plans:
Receives and Restore plans
Apply events suspended during this phase*

backup mode run

*Receives and applies events
to plans and data*



Synchronization - plans extension (CP or LTP)

transfer plans

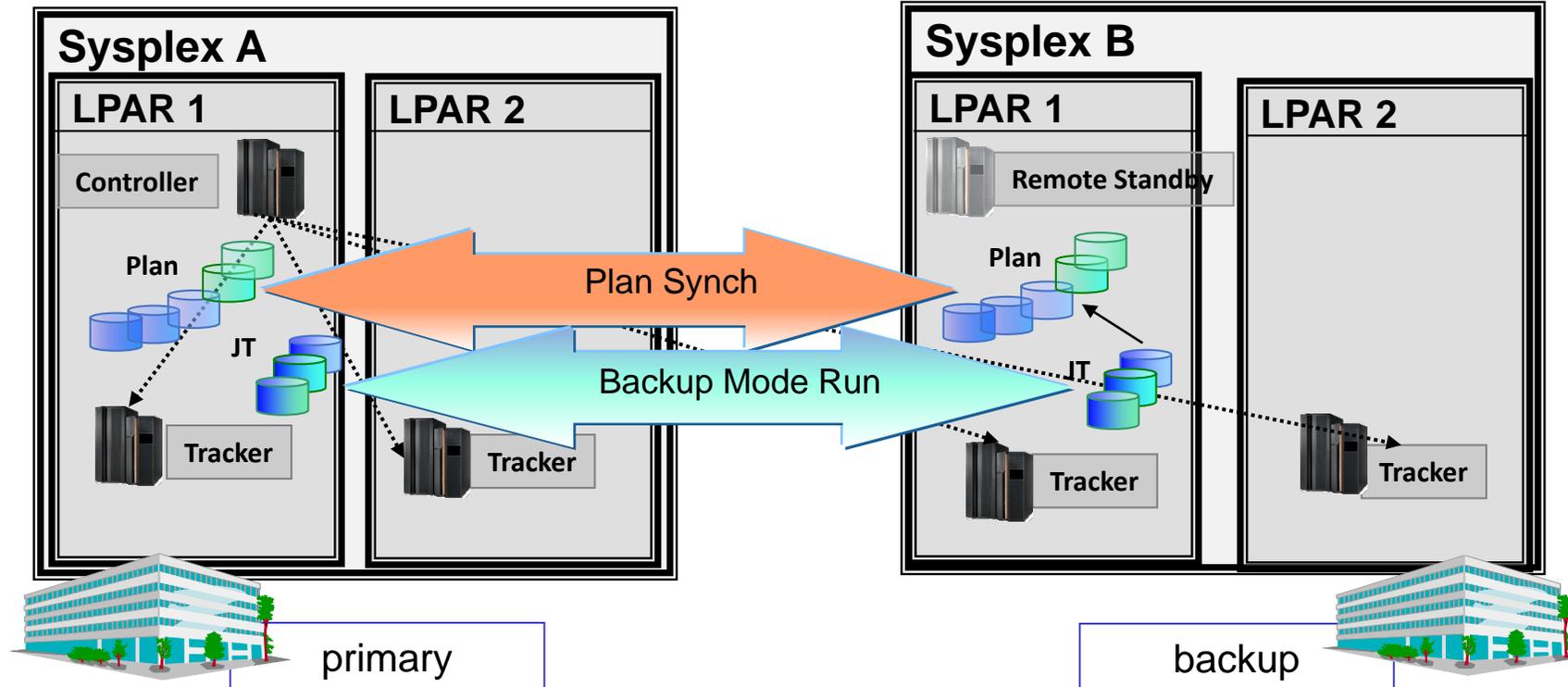
*primary Controller
transfers
plans to backup via FTP*

restore plans

*backup Controller starts the plan
synchronization:
Receives and Restore plans
Apply events suspended during
this phase*

backup mode run

*Receives and applies events
to plans and data*



BKTAKEOVER command: options

You can choose different ways to make the backup controller take up the functions of primary Controller



/f subsys_name BKTAKEOVER

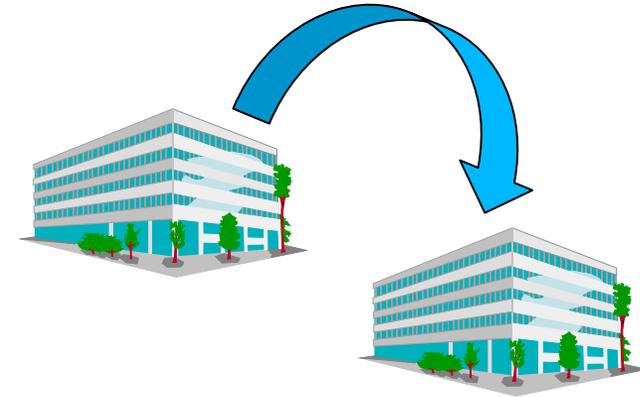
This command, issued on a backup Controller, orders it to take over the functions of the primary Controller. You can use additional options of this command :

- ◆ Specify the option **BKTAKEOVER,FORCE** to force the backup Controller takeover, even when the connection between the controllers seems to be up and running.
- ◆ Specify the option **BKTAKEOVER,NOSUB** to deactivate the job-submit option, even if JTOPTS JOBSUBMIT(YES) is set.

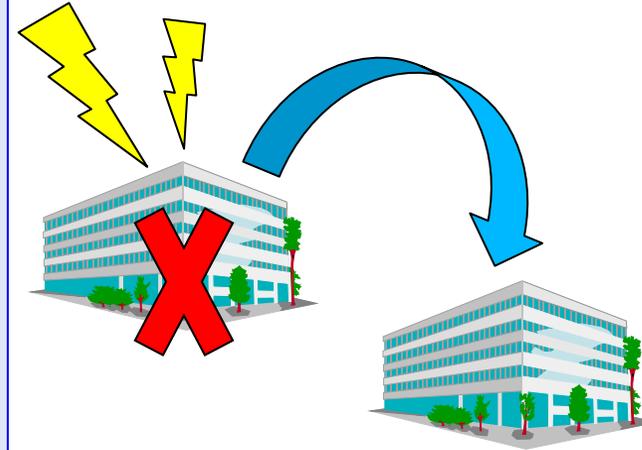
- 1) Execute a CP backup, than submit the JS transfer procedure for the inactive JS
- 2) Stop the primary controller
- 3) On backup Controller MLOG wait for the message about the connection status:

EQQBT17W CONNECTION RESET BY PEER

- 4) At this point issue the modify command **BKTAKEOVER**
- 5) The available trackers and remote engines automatically establish a connection and synchronize with the backup Controller



- 1) On the backup Controller, enter the modify command **BKTAKEOVER**
- 2) The backup Controller applies all the events in queue and switches from normal backup mode to primary mode.
- 3) The connection with the Trackers is automatically established and a synchronization process occurs.
- 4) Check the jobs that were in started status on the primary Controller when the connection failed, and manually correct the status on the backup Controller



New filter:

STARTED AT STARTUP

A new filter can be used to list only the operations that were in started status when the Controller starts, either the primary or the backup after the take over.

Warning and informational messages are issued on backup Controller to indicate that manual actions are required in case of failover occurred during a previous synchronization phase

Restoring the primary Controller

When you want to run the reverse processing to restore the primary Controller:



To switch back from the backup Controller to the primary Controller:

- ◆ On the Controller that you want to restore as the primary, set **OPCOPTS** parameter **OPCHOST(BACKUP)** and start it as the backup Controller. The current backup Controller and primary Controller will start the synchronization phase.
- ◆ Stop the current primary Controller.
- ◆ On the backup Controller, issue the modify command **BKTAKEOVER**. In this way, the backup Controller takes over the functions of the controlling system.
- ◆ Start the Controller that you stopped in step 2 as the backup Controller

Remote Sites Timezones Considerations



If the primary Controller and the backup Controller run in different time zones, to ensure that the backup Controller scheduling respects the time dependencies according to the time of the primary Controller, customize the time-dependent-operation exit (**EQQUX014**) appropriately.