



IMS Disaster Recovery

Part 2 - iRecover and iChange

William N. Keene
NEON Enterprise Software, Inc

Tuesday, April 8, 2008





Author

Bill Keene has almost four decades of IMS experience and is recognized world wide as an expert in IMS recovery and availability. Prior to retiring from IBM, Bill was the technical leader of the IMS unit within the Dallas Systems Center, part of IBM's Advanced Technical Support organization. In that capacity, Bill presented at numerous IBM and user technical conferences and was the technical focal point within IBM's marketing division for product introduction programs dealing with XRF, RRDF, and RSR, all IMS functions dealing with local system or disaster recovery.

Bill retired from IBM in 1998 and is currently utilizing his IMS database recovery knowledge and expertise as the product architect of iRecover and iChange, the primary components in the IMS database recovery solutions provided by NEON Enterprise Software, Inc.





Agenda

Disaster Recovery Basics

Eclipse iRecover

Eclipse iChange

Q & A



Disaster Recovery Basics

Objective:

- Restore IMS service at a remote location when the local (normal) computing facility is no longer available
- Remoteness is in the eye of the beholder
 - In general, the further away, the better
- Ensure the survival of the business!



Disaster Recovery Basics . . .

Disaster Recovery Methodology

- Determined by business needs
 - How much data can business lose?
 - How long can business go without service?
 - What can business afford?
 - Is DR methodology allowed to impact local availability
- Cost goes up as data loss and service unavailability go down



Areas of Opportunity

Pick list creation

- Need to know what to send off-site

RECON backup / restore

- Spanned / segmented record considerations

RECON cleanup

- Time consuming, cumbersome, and error prone manual operation

Database recovery

- Need to recover thousands of database data sets / AREAs quickly, correctly, and to a consistent state; and then rebuild indexes, perform hash pointer checking, and take image copies



Types of Recovery

Full recovery

- Recover object to its most current state

Time Stamp Recovery

- Recover object to a prior state when no updates in progress

Point-In-Time Recovery

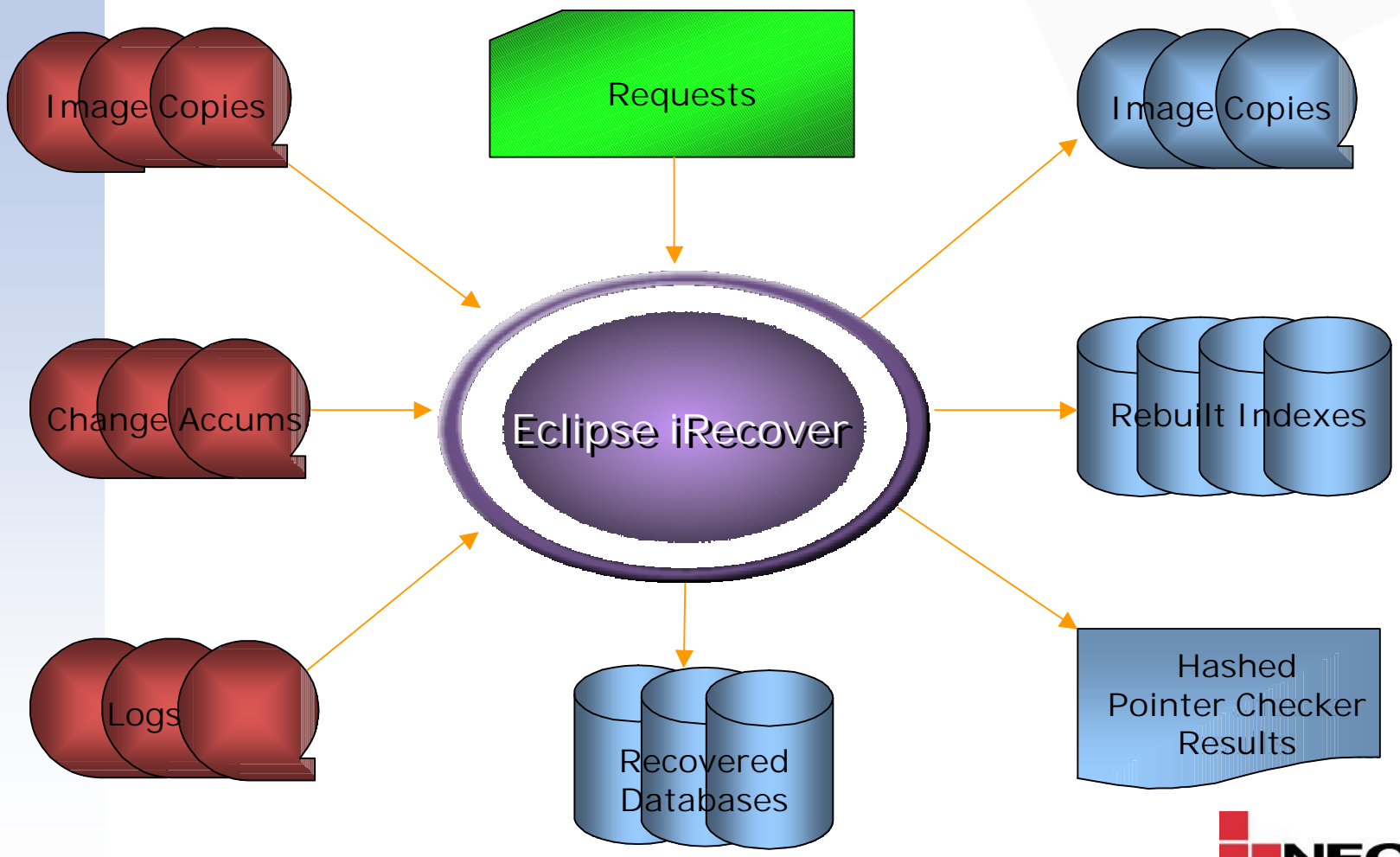
- Recover object to a prior, consistent state when updates in progress

Disaster Recovery at a remote recovery site

- Recover the world easily



Eclipse iRecover Architecture





Eclipse iRecover

Determines and allocates all inputs and outputs from information contained in the RECONS and control statements

```
//RECOVER JOB . . .
//STEP EXEC NSRMAIN[ ,PARM='option-mbr-name' ]
//SYSPRINT DD SYSOUT=* (optional)
//SYSUDUMP DD SYSOUT=*
//OPTIONS DD * (optional)
    One or more SETOPTIONS statements
/*
//NEONIN DD *
    One or more recovery control statements
/*
```



Eclipse iRecover . . .

Recovers thousands of database data sets efficiently in a single operation

- Application failures and media failures require that multiple database data sets be recovered. Disaster recovery requires that the world be recovered.
- User sets number of concurrent processes
- Reduces data outage
- Example
 - RECOVERGRP DBD(*)



Eclipse iRecover . . .

Recovers all types of IMS databases using common procedures and commands

- IMS knows if a database is Full Function or Fast Path and whether it has any indexes. Why should you?
- Simplified recovery operations
- Recovery commands
 - RECOVERDBDS DBD(*name*) DD(*name*)
 - RECOVERGRP DBD(*generic name*)|GROUP(*name*)|CAGROUP(*name*)|PSB(*name*)
 - RECOVERVOL VOLSER(*list*)
 - RESTOREDBDS DBD(*name*) DD(*name*) DSN(*name*) ICDSN(*name*)



iRecover . . .

Recovers database data sets to any point-in-time (pick a time, any time)

- No need to reduce data availability by creating recovery points
- Provides unlimited options to the user when the recovery is in response to an application failure
- You, rather than the recovery utility, are in control
- Allows for coordinated application recovery with DB2
- Example
 - RECOVERGRP GROUP(PAYROLL) -
RECOVERTO(2008.057 10:23:14.123456)



iRecover . . .

Validates pointers and FSEs while recovering

- Eliminates need to perform hashed pointer check following recovery
- No elapsed time cost
- Nice, warm, fuzzy feeling
- Reduces data outage
- Each block/CI checked structurally
- Optional when basis of recovery is a DFSMS-type backup (IC2, USEDBDS, etc.)



iRecover . . .

Rebuilds indexes while recovering

- Eliminates need to schedule index rebuild jobs following recoveries
- Rebuild, recover, iBuild? Your choice!
- Reduces data outage
- Index rebuild occurs in subordinate address spaces
- Rebuild starts once all targets have been recovered
- User sets number of concurrent index rebuild threads
- Example
 - RECOVERGRP GROUP(PAYROLL) IB(YES|RECOVER
IFNEEDED|FORCE) IBUILD(SOME)



iRecover . . .

Creates new image copies while recovering

- Eliminates need to take image copies after recovering
- Reduces impact of subsequent recoveries
- Reduces data outage
- Number of image copies and allocation information defaults to input image copy characteristics
- Pattern data set names supported
- Examples
 - RECOVERGRP GROUP(PAYROLL) ICOUT(YES)
 - RECOVERGRP GROUP(SKILLS) ICOUT(2)



iRecover . . .

Recovers only those database data sets that need recovering

- You state your objectives – Eclipse iRecover figures out how to do it
- No recovery is faster than the recovery you don't do
- Reduces outage
- Object needs recovering if
 - Shadow database data set, 'Recovery needed' flag on, or ALLOC exists beyond recover to time
- Example
 - RECOVERGRP GROUP(PAYROLL) - RECOVER(IFNEEDED)



iRecover . . .

Recovers only those indexes that need rebuilding

- No rebuild is faster than the rebuild you don't do
- Reduces outage
- Index needs rebuilding if
 - Shadow index, 'Recovery needed' flag on, or ALLOC exists beyond recover to time for target
- Example
 - RECOVERGRP GROUP(PAYROLL) –
IB(YES IFNEEDED)



iRecover . . .

Recovers related database data sets automatically

- Maintain physical integrity and consistency when PIT recoveries performed
- Protects against user error!
- Based on DBD defined relationships
- Examples:
 - RECOVERGRP DBD(INVENTORY) RELATED(YES)
 - RECOVERGRP DBD(SKILLS) RELATED(FORCE)



iRecover . . .

Supports shadow database recoveries

- Unregistered database copies for testing, reporting, etc.
- Complete PIT support
- Full Function, Fast Path, and HALDB
- Pattern data set names supported
- Fast Path DMAC CI can be changed during recovery
- Examples:
 - SETOPTIONS SHADOWDSNHLQ(TEST1)
 - RECOVERDBDS DBD(PAY1) DD(PAY1) DSN(TEST1.PAY1, TEST2.PAY1)
 - RECOVERGRP GROUP(PAYROLL) DSN('TEST1.&DB.&DS')



iRecover . . .

Supports data set name patterning

- Multiple levels of patterning
 - ICDSN – Specifies pattern for output image copies
 - DSN – Specifies pattern for output database data sets
 - FPDSN – Specifies over-ride pattern for FP database data sets
 - HALDSN – Specifies over-ride for HALDB database data sets
 - NDXDSN – Specifies over-ride for index database data sets
 - NDXDSNH – Specifies over-ride for PSINDEX database data sets
 - NDXDSNS – Specifies over-ride for secondary index database data sets



iRecover . . .

HALDB considerations

- Master names or partition names can be specified in RECOVERGRP DBD(*name*) commands
- Time stamp and PIT recovery supported
 - Correct set of data sets determined when RECOVERGRP or RECOVERVOL commands used (A-J or M-V)
 - Correct set not necessarily current set
 - User must specify correct data set when RECOVERDBDS command used
 - PIT into middle of OLR presents challenges, but is supported
- RELATED(YES) processing for a partition will cause all partitions to be considered for processing



iRecover . . .

HALDB considerations

- Index rebuild
 - PSINDEX rebuild currently requires use of iBuild support
 - Primary index rebuild is supported
 - ILDS rebuild is supported
- Shadow database recovery is supported but
 - Getting a shadow RECON in-sync with the shadow HALDB database datasets is the user's responsibility
 - HALDB Partition Definition Utility EXPORT/IMPORT



iRecover . . .

Takes advantage of existing change accums, if available

- Less log data to read and process
- Reduces data outage
- Change accum data extracted while log and IC2 image copies are being processed

A change accum can be used with PIT recovery as long as it does not contain any logs that ended after the requested PIT



iRecover . . .

Non-standard backups (User Image Copies) treated as fuzzy

- No problem treating a consistent pack dump as fuzzy
- Major problems treating a fuzzy pack dump as consistent
- When in doubt, perform a correct recovery!
- iRecover does not contain any magic to handle KSDS splits that might occur while a pack dump is performed



iRecover . . .

Audit function

- Determines what to do to whom with what
- Reports same
- Optional recovery inventory report
- Example
 - SETOPTIONS AUDIT(YES) INVDSN(*)
 - RECOVERGRP DBD(*)



iRecover . . .

Recovery simulation function

- Performs complete recovery process (read and process all inputs), but does not recover to DASD
- Validate ability to recover without requiring duplicate DASD
- Typical use would be a disaster recovery audit
- Example
 - SETOPTIONS RECOVER(NO)
 - RECOVERGRP DBD(*)



iRecover . . .

The following processes all operate in parallel

- Multiple log processing threads
- Multiple IC2 image copy restore threads
- Multiple CA data set processing threads
- Multiple recovery threads
- Multiple index rebuild threads and address spaces

User specifies the number of threads (tasks) for each process



iRecover . . .

AMS DELETE/DEFINE, DELETE/ALLOCATE support

- Multiple levels of support (EXISTING,PDSDEF, CATDEF, or NONE)
- Choice of DELETE SCRATCH or NOSCRATCH
- Both DBD and DD member names supported
- Members are searched for IDCAMS statement containing the data set name or cluster name being processed
 - Allows same library to support multiple systems
- Any DELETE statements are ignored



iRecover . . .

Processing modes

- SCAN
 - All blocks / CIs processed by iRecover recovery thread
 - All iRecover functions supported
 - Associated with, but not limited to, classic image copy recoveries (default)

- SKIP
 - Only those blocks / CIs that have change accum and/or log updates processed by iRecover recovery thread
 - iRecover index rebuild, hash checking, and output image copy functions disabled
 - Associated with, but not limited to, IC2 and USEDBDS recoveries (default)



iRecover . . .

Provides enhanced incremental image copy function

- Change accumulation not required
- Consistent batch or concurrent image copies can be created to any PIT
 - Contents of the image copies will be consistent
- Can create concurrent image copies for VSAM KSDS
- Example
 - RECOVERGRP GROUP(PAYROLL) -
RECOVER(INCRIC) RECOVERTO(LASTLOG)



iRecover . . .

Scenario

User runs a BMP that corrupts the databases at T2. He discovers the error and takes the databases off-line at time T3. The latest image copy or change accum is at T1.

40 logs were created between T1 and T2. 2 logs were created between T2 and T3.

Wouldn't it be nice to be able to backout the databases from their current state (T3) to T2, rather than perform a PIT recovery to T2?



iRecover . . .

iRecover V6 contains a new command, PRGREVOCER. It also has an alias, BACKOUTGRP.

BACKOUTGRP allows the user to perform BACKOUT processing against a group of database data sets to a user specified time stamp

The backout operation maintains transactional consistency

Example

- BACKOUTGRP GROUP(PAYROLL)
BACKOUTTO(2008.057 13:00:00)



iRecover . . .

BACKOUTGRP considerations

- Restrictions
 - DEDBs not supported
 - Database data sets must not need recovering
 - Non-recoverable databases not supported
 - Not allowed if any partial (time stamp or PIT) recoveries have been performed after the BACKOUTTO time stamp
 - Concept of shadow copies not supported
- BACKOUTGRP BACKOUTTO(time n) and RECOVERGRP RECOVERTO(time n) will accomplish the same result



iRecover . . .

ISPF front-end

- Assists casual user with control statement construction
- Extensive help panel support
- Reduces data outage

Supports NESI Mission Control Repository

- Can specify repository name, if desired
- Can specify MC load library name, if desired
- User controls how and what to access
- Example
 - SETOPTIONS MCLIB(*name*) MCREPOS(*name*) –
MCUNIQUE(YES|NO) MCUSE(YES|NO)



iRecover DR support

RECON Cleanup Utility

- Creates “Load and Go” RECON backup conditioned for DR
 - All necessary cleanup operations already performed
- RECON analyzed to determine DR time stamp validity
 - User specified time validated or latest possible DR time determined
- Recovery media inventory report
 - Can be used as a “pick list” to send stuff off-site
 - Can be used to catalog data sets at DR site
- Latest Eclipse iChange time stamp



iRecover DR Support . . .

Log Analysis Utility

- Analyzes all logs (SLDS and batch) spanning a user-specified time stamp
- Reports last symbolic checkpoint id for all DLI/DBB and BMP programs that were executing at the reporting time
- Lists status (commit or abort) of all ESAF (DB2) UORs that were in-doubt at the reporting time



DR Recovery Steps

Restore latest DR RECON backup

```
//DR1      JOB      . . .  
//STEP1    EXEC     PGM=NSRRCNRS  
//STEPLIB  DD       DSN= . . .  
//BACKUP   DD       DSN= . . .  
//RECON1   DD       DSN= . . .
```

Restore RECON duality

```
//DR2      JOB      . . .  
//STEP1    EXEC     PGM=DSPURX00  
//STEPLIB  DD       DSN= . . .  
//SYSPRINT DD       SYSOUT=*  
//SYSIN    DD       *  
          LIST.RECON STATUS  
/*
```



DR Recovery Steps

Run Eclipse iRecover (recover to 8 AM Monday)

```
//DR3      JOB      . . .  
//STEP1    EXEC    PGM=NSRMAIN  
//STEPLIB DD      DSN= . . .  
//NEONIN   DD      *  
            RECOVERGRP DBD(*) -  
            RECOVERTO(2008.056 08:00:00)  
/*
```

Go get coffee!



AMSPDS Build Utility

Utility provided to build IDCAMS control statements for registered and shadow database data sets and clusters

Two pass architecture

- Extract allocation information for registered database data sets and clusters and store it in a repository (KSDS – 24 bytes keys)
- Build IDCAMS control statements for registered or shadow database data sets and clusters

Common look and feel with rest of iRecover

- Groups and related processing supported
- Shadow pattern data set names supported



AMSPDS Build Utility . . .

Control Statements

- SETOPTIONS
- EXTRACTDBDS
- EXTRACTGRP
- BUILDDBDS
- BUILDGRP
- DATACLAS
- STORCLAS

Special HALDB OLR support



Recovery Analysis Utility

Analyzes information in RECONS and MVS catalog

Purpose is to identify conditions that could

- Prevent a database data set from being recovered
- Elongate the recovery of a database data set

Analysis can be driven from a database data set perspective and/or a log perspective

Example

```
— ANALYZE DBD(name)|CAGROUP(name)|  
GROUP(name)|LOGS
```



Database Analysis Conditions

Always checked

- BACKOUT NEEDED
- CA in error
- IC in error
- IC NEEDED / RECOMMENDED
- No available ADS
- PINIT required
- Disabled
- Open cursor
- READ ONLY
- RECOVERY NEEDED
- PROHIBIT AUTHORIZATION



Database Analysis Conditions . . .

User requested

- ALLOC age (threshold)
- CA age (threshold)
- CA not cataloged (threshold)
- IC age (threshold)
- IC not cataloged (threshold)
- Max logs for recovery (threshold)
- Missing recovery assests
- Not in a CA group
- No secondary image copy
- Unavailable ADS



Log Analysis Conditions

Always checked

- Inactive logs
- Logs in error
- Logs open
- Archiving gaps
- Archive needed (threshold)

User requested

- Log not cataloged (threshold)
- No secondary log (RLDS)
- No secondary log (SLDS)
- Log data set age (threshold)



Eclipse iChange

iChange



Eclipse iChange

Determines and allocates all inputs and outputs from information contained in the RECONS and control statements

```
//RECOVER  JOB      . . .  
//STEP     EXEC    NSAMAIN[ ,PARM='option-mbr-name' ]  
//SYSPRINT DD      SYSOUT=* (optional)  
//SYSUDUMP DD      SYSOUT=*  
//OPTIONS  DD      *          (optional)  
           One or more SETOPTIONS statements  
/*  
//NEONIN   DD      *  
           One or more ACCUMGRP control statements  
/*
```



Eclipse iChange . . .

Normal change accumulation processing

- Multiple CAGRPs processed in a single pass of the logs
 - Generic group names supported
- Logs, including FP logs, read in parallel
 - User specifies level of parallelism
- CAGRPs processed in parallel
 - Users specifies level of parallelism
- Each GAGRP has its own SORT address space
 - Automatically started and stopped
- Extensive encoding used to reduce amount of data sorted



Eclipse iChange . . .

Target disaster recovery environment

- User needs only a few DR points per day
 - Recovering to prior 2 AM is good enough
- User does not want to quiesce database update activity or data sharing level when establishing a DR point
- User wishes to send only image copy and change accum data sets off-site (no logs), along with a RECON backup
- User wishes to recover databases to a consistent state (prior 2 AM) at the disaster site
 - Quickly, correctly, with minimum effort



Eclipse iChange . . .

Disaster recovery support ...

- Special DR change accumulation data sets
 - Normal change accumulation information
 - Additional records, if needed, to restore consistency
- “Load and Go” RECON backup conditioned for DR
 - All necessary “cleanup” operations already performed
- DR recovery media inventory report
 - Can be used to create “pick list” of data sets to go off-site
 - Can be used to catalog data sets at DR site
- “DR Coverage” report
 - Lists all registered recoverable database data sets that are not in a DR change accumulation data set
 - Protects against missing DR coverage



Eclipse iChange . . .

Steps to establish a 2:00 AM DR point at the local site

- Automation issues /SWI OLDS immediately after 2:00 AM
- Automation submits Eclipse iChange job after archives complete

```
//DRCA      JOB      . . .
//STEP1     EXEC     PGM=NSAMAIN
//STEPLIB   DD       DSN= . . .
//NEONIN    DD       *
              ACCUMGRP GROUP(CAGRP1) ACCUMTO(* 02:00:00)
              . . .
              ACCUMGRP GROUP(CAGRP2) ACCUMTO(* 02:00:00)
              . . .
/*
```

- DR change accums and RECON backup sent off-site, along with any image copy data sets created since last DR point



Eclipse iChange . . .

Steps to accomplish disaster recovery

- Restore latest DR RECON backup

```
//DRCA      JOB      . . .  
//STEP1    EXEC     PGM=NSARCNRS  
//STEPLIB  DD       DSN= . . .  
//BACKUP   DD       DSN= . . .  
//RECON1   DD       DSN= . . .
```

- Restore RECON duality

```
//DR2       JOB      . . .  
//STEP1    EXEC     PGM=DSPURX00  
//STEPLIB  DD       DSN= . . .  
//SYSPRINT DD       SYSOUT=*  
//SYSIN    DD       *  
LIST.RECON STATUS  
/*
```



Eclipse iChange . . .

Steps to accomplish disaster recovery

- Run Eclipse iRecover

```
//DRCA      JOB      . . .  
//STEP1     EXEC     PGM=NSRMAIN  
//STEPLIB   DD       DSN= . . .  
//NEONIN    DD       *  
            RECOVERGRP GROUP(CAGRP1)  
            RECOVERGRP GROUP(CAGRP2)  
/*
```

- Go get coffee!



Coordinated Recovery (DB2)

Local recovery

- Recover DB2 objects to desired state (typically a DB2 QUISCE point)
- Recover IMS databases to same state using PIT time stamp (DB2TIME YES) or LRSN value

Disaster recovery (IMS log input available)

- Perform conditional DB2 restart to user desired state (standard DB2 facility)
- Recover IMS databases to same state using PIT time stamp (DB2TIME YES) or LRSN value



Coordinated Recovery (DB2)

Disaster recovery (Eclipse iChange DR CA available)

- Perform conditional DB2 restart to LRSN associated with the Eclipse iChange DR change accum (standard DB2 facility)
- Recover IMS databases to same state using DR RECON created by Eclipse iChange



Common Tasks

Create three shadow copies of database data set to a 2 AM consistent state

- RECOVERDBDS (DBD(DBDA) DD(DBDA) –
RECOVERTO(* 2:00:00) DSN(DSN1 DSN2 DSN3)

Recover all database data sets that reside on volumes IMS001 and IMS002 to their most current state

- RECOVERVOL (VOLSER(IMS001 IMS002))

Recover all of the data sets in database DBDX to their latest valid batch image copy

- RECOVERGRP (DBD(DBDX)
RECOVERTO(LASTIC))



Common Tasks . . .

Recover all database data sets belonging to group APPL1, along with all related database data sets, to 8 AM today, but only perform recoveries if really needed

- `RECOVERGRP (GROUP(APPL1) RELATED(YES) – RECOVER(IFNEEDED) RECOVERTO(* 8:00:00))`

Recover all database data sets belonging to group APPL2, along with all related database data sets, to a time stamp represented by the LRSN reported by a DB2 QUIESCE

- `RECOVERGRP (GROUP(APPL2) RELATED(YES) – RECOVERTO(LRSN(xxxxxxxxxxxx)))`



Common Tasks . . .

Recover all database data sets to a consistent state as of 10:00:00 GMT yesterday

- RECOVERGRP (DBD(*) RECOVERTO(2008.056 10:00:00 GMT))

Create incremental image copies, at a 7:00 AM state, for all databases data sets in group APPL1

- RECOVERGRP (GROUP(APPL1) – RECOVERTO(* 7:00:00) RECOVER(INCRIC))

Recover DBDS to its most current state using a fuzzy pack dump taken at 2:00 AM Sunday as the basis of recovery

- RECOVERGRP (DBD(DBDX) RELATED(NO) – RECOVER(USEDDBDS) – RECOVERFROM(2008.055 2:00:00))



Common Tasks . . .

Determine what would be needed to recover all databases starting with PAY and related database data sets, to 8 AM today, but just those that really need to be recovered

- SETOPTIONS AUDIT(YES)
RECOVERGRP (DBD(PAY*) RELATED(YES) -
RECOVER(IFNEEDED) RECOVERTO(* 8:00:00))

Go through all the steps required to recover all registered recoverable databases, including indexes, to 8 AM today, but don't actually recover the databases

- RECOVERGRP (DBD(*) IB(RECOVER FORCE) –
RECOVER(NO) RECOVERTO(* 8:00:00))



Q & A

Questions?