Technical White Paper

A Coordinated Recovery Process

Obtaining Outage Free Recovery Points for IMS and Db2 Disaster Recovery

Table of Contents

Executive Summary

Many large organizations started with IMS databases and later added Db2 into their environments. As applications were developed and updated, they may be accessing both IMS and Db2 data or an application in one environment may spawn an update in the other. These types of situations require that recovery be coordinated so that both IMS and Db2 are restored to a common point in time to avoid data corruption and loss.

BMC recovery solutions provide a way to find points of consistency between IMS and Db2 and automate the recovery of those environments to that common point of consistency. This process can be used for both disaster recovery situations and local recoveries.

Read on to learn more about:

- Finding points of consistency
- Outage free coordinated recovery
- A process to follow to implement coordinated IMS and Db2 recovery

THE REQUIREMENT FOR COORDINATED RECOVERY

Management

In many large data processing organizations, applications have evolved and established relationships between IMS databases and Db2 databases. Sometimes these relationships are known to the application (as in the case of a two-phase commit process), but frequently it's a more casual relationship. A transaction in one database management system may spawn a subsequent transaction in a different database management system. The transactions are tied, but the log data isn't coordinated.

This situation requires a coordinated recovery between IMS and Db2 applications for both local recovery and disaster recovery. To ensure data integrity, transactions and batch jobs on all systems are drained or stopped to establish a system- wide point of consistency (SWPOC). This technique is preferable from a data integrity standpoint, but it causes an availability impact that is in conflict with a 24X7 environment.

Outage-Free Coordinated Recovery

Using exclusive BMC Software technology, it's possible to capture information from the database repositories and use it to derive a consistent and coordinated recovery point. The probable use for this process is in support of disaster recovery (DR), but some applications (for example, large ERP applications with correlated legacy applications) may make use of the procedure for local recoveries.

The following sections explain some examples of exploiting BMC Software technologies to create a coordinated recovery point. The first section describes the process of taking an IMS log switch timestamp and using it to create a recovery point at that timestamp for both IMS and Db2 applications. The final section contains some REXX code to automate some of the operations.

Note: Use the processes described in this paper as examples. Your shop will likely have standards or procedures that will require some local modifications to these processes.

A COORDINATED RECOVERY PROCESS FOR IMS AND DB2

The following section describes a coordinated recovery process for use in IMS and Db2 shared environments.

Overview

The coordinated recovery process for IMS and Db2 consists of the following major tasks:

1. Use the BMC Software RECOVERY MANAGER (RMGR) for IMS Disaster Recovery RECON Cleanup (DRRCN) utility to report on the RECONs, looking for the latest, safest timestamp that can be used in a Point-In-Time (PIT) recovery. This timestamp is the STOPTIME of the oldest SLDS (the one with the least recent timestamp) in an IMS data-sharing environment. RMGR for IMS ensures that all open logs are properly handled before choosing the PIT timestamp.
2. Pass the captured timestamp to the RMGR for Db2 Timestamp Insertion (ARMBTSI) batch program, which inserts the timestamp into the RMGR CRRDRPT repository table.
3. Execute the RMGR for Db2 Coordinated Disaster Recovery (ARMBCRC) batch program, which formats the timestamp into a relative byte address (RBA) or, in a Db2 data-sharing environment, a log record sequence number (LRSN). The RBA or LRSN is used to generate a Db2 conditional restart control record (CRCR) to that RBA or LRSN.
4. Execute the RMGR for Db2 System Resource Recovery (ARMBSRR) batch program, which generates a Db2 subsystem recovery and restart based on the CRCR RBA or LRSN that was built by the ARMBCRC program in the prior step. The Db2 restart process backs out any in-flight transactions at the RBA or LRSN.
5. Pass the latest IMS timestamp to RMGR for IMS to generate one or more RECOVERY PLUS for IMS jobs that recover the databases to this timestamp. Transactions that were in-flight at the timestamp are not applied during the recovery.

Note: This process assumes that the IMS and Db2 systems share a sysplex timer; otherwise, the timestamps will not coincide.

The net effect to your application is just as if the power was dropped at the local site at the time represented by the PIT timestamp and you restarted all the IMS and Db2 subsystems. You'd lose the work that was in-flight at the time of the power failure. You'd be at the same position at the DR site. You can use the RMGR for IMS Log Analysis function to obtain a report of work that was in progress at the timestamp you used for recovery.

The process flow includes application copies before the coordinated recovery process and dumps of the appropriate system files (catalogs, directories, libraries, and so on) after the coordinated recovery process.

The Details

The following sections explain the coordinated recovery process for IMS and Db2 in detail.

Preparation Steps

Perform the following preparation steps:

Step	Action	Description
1	Prepare the DR site.	Ensure that valid copies of the IMS and Db2 system libraries and BMC Software RMGR product libraries are available at the DR site.
2	Create and validate RMGR groups.	Create and populate RMGR groups to be used for possible DR scenarios. Use the Group Validation function regularly to ensure that groups are kept updated with changes in database allocations.
3	Create an RMGR profile for DR.	Create an RMGR for IMS DR profile that is tailored for usage at the DR site (for example, turn on secondary image copies and secondary logs for use in recovery, check output devices that are used for change accumulation, review SORT parameters, and so on).
4	Check recovery assets.	Periodically (perhaps monthly) verify the DR process by performing the Check Assets function (to ensure availability of recovery assets) and by using the Create Recovery JCL function (to create practice JCL).

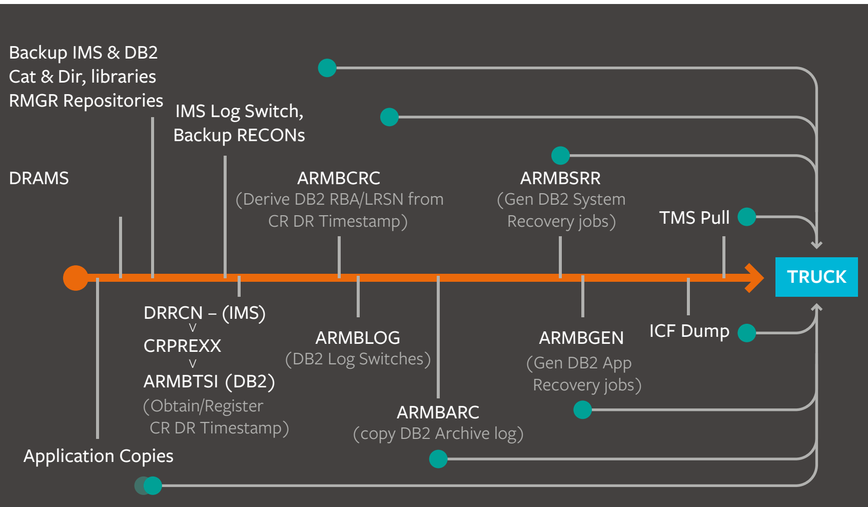
At the Local Site

On a periodic basis (perhaps nightly), perform the following preparation steps:

Step	Action	Description
1	Copy objects.	Copy the IMS and Db2 application objects. (You can create clean or fuzzy copies; it doesn't matter.)
2	Capture allocation information.	Run the RMGR for IMS Automatic Delete/Define (DRAMS) utility Capture function to store allocation information about database data sets in the RMGR for IMS repository.
3	Back up system objects.	Back up the IMS RECONs, Db2 catalog and directories, all BMC Software RMGR product repositories, and all IMS and Db2 system libraries and data sets. Prepare these backups for shipment to the remote site.
4	Create system log data sets.	Switch the IMS online log data sets (OLDSs) to create system log data sets (SLDSs) on tape, and prepare them for shipment.
5	Generate the DR timestamp.	Execute the RMGR for IMS RECON Cleanup (DRRCN) utility Check function to generate a report that contains the LATEST PIT RECOVERY TIME IS: value field. The value is the coordinated recovery disaster recovery timestamp (CR DR TS).
6	Conduct date/time conversion.	Invoke the CRPREXX REXX EXEC program (provided in this document) to browse the DRRCN report output and conduct date/time conversion on the CR DR TS. Capture the timestamp value and ship it to the remote site with the other recovery assets.

7	Pass the converted timestamp value to ARMBTSI	Pass the converted CR DR TS value to the RMGR for Db2 Timestamp Insertion (ARMBTSI) batch program, which inserts a row into the CRRDRPT repository table.
8	Translate the timestamp to an RBA/ LRSN value.	Invoke the RMGR for Db2 Coordinated Disaster Recovery (ARMBCRC) batch program, which takes the timestamp value from the CRRDRPT table and translates it into the proper RBA/ LRSN value. This value becomes the input to the Conditional Restart Control Record (CRCR) process that is used later by the RMGR for Db2 System Resource Recovery (ARMBSRR) batch program.
9	Archive the active Db2 logs.	Archive the active Db2 logs through the RMGR for Db2 Archive Log Creation (ARMBLOG) batch program.
10	Copy the archive logs.	Invoke the RMGR for Db2 Archive Log Data Sets (ARMBARC) batch program to copy the logs that are needed for recovery, and ship the copies to the remote site. ARMBARC also serves the very useful purpose of capturing the image copy information for the three "special" Db2 Catalog/Directory objects (DSNDB01.DBD01, DSNDB01.SYSUTILX, and DSNDB06.SYSCOPY. These objects are fundamental to the success of Db2 Disaster Recovery, and image copy information is not stored in the normal fashion on DSNDB06.SYSCOPY.
11	Generate JCL to rebuild Db2 at the remote site.	Pass the CRCR RBA/LRSN value to the ARMBSRR batch program, which generates the 200+ job steps to rebuild Db2 at a remote site to the designated RBA/LRSN (the CRCR established by the ARMBCRC program). The ARMBSRR program also generates recoveries for the RMGR for Db2 repository tables. Ship the output of the ARMBSRR program to the remote site.
12	Generate JCL to recover application data.	Invoke the RMGR for Db2 Backup and Recovery JCL (ARMBGEN) batch program to generate recovery JCL for the application data. Ship the output to the remote site.
13	Back up the ICF catalog and TMS directory.	Back up the operating system catalog (ICF) and the Tape Management System (TMS) directory, and prepare the backups for shipment to the remote site.

Figure 1 - The Outage Free Coordinated Recovery Point Capture Process (Local Site)



At the Remote Site

When you are performing disaster recovery at the remote site, the process is relatively straightforward. After the operating system has been restored, you simply restore the IMS RECONs and prepare them for a cold-start of IMS, release the jobs that are generated by the ARMBSRR and ARMBGEN programs, and generate IMS database recoveries to recover the data to the CR DR TS that was captured by the DRRCN utility and that was used by the CRPREXX program. Any in-flight IMS or Db2 transactions that were active at the time of the CR DR TS are not recovered.

Perform the following steps:

Step	Action	Description
1	Restore IMS system and product libraries.	Restore IMS RECON (RECON1) and system libraries, as well as the RMGR for IMS data sets.
2	Clean up the RECON.	Execute the RMGR for IMS RECON Cleanup (DRRCN) utility Update function to prepare the IMS RECON data sets for startup. Make a note of the value that the utility reports in the LATEST PIT RECOVERY TIME IS: value field.
3	Allocate and REPRO RECONs.	Allocate RECON2 and RECON3 data sets, and REPRO the cleaned RECON1 to RECON2.
4	Perform IMS startup.	Cold-start IMS, and start RMGR for IMS.
5	Perform IMS change accumulation (optional).	Perform IMS change accumulation if you want to do so. Refer to the RECOVERY PLUS for IMS documentation for caveats and recommendations for timestamp selection.
6	Generate PIT recovery JCL for IMS objects.	Use the value from the LATEST PIT RECOVERY TIME IS: value field (from the execution of the RECON Cleanup utility at the local site); this value is also the CR DR TS that is used in the CRPREXX program. Enter this value in the RMGR for IMS ISPF dialog to generate the appropriate RECOVERY PLUS for IMS JCL to conduct a PIT recovery to that timestamp.
7	Release PIT recovery JCL.	Release the generated IMS PIT recovery jobs (the PIT timestamp is the time that was captured in the RECON Cleanup utility previously in this procedure).
8	Recover the Db2 system.	Execute the jobs that the ARMBSRR program generates to recover the Db2 system and RMGR for Db2 repository. (The two jobs are stored in one data set). Remember to specify the DEFER ALL parameter in the ZPARMS for the remote site startup; if you do not, disk allocation failures occur as Db2 tries to mount volumes to back out in-flight transactions on data that doesn't exist ... yet.
9	Recover Db2 applications.	Release the Db2 application recovery jobs that are generated by the ARMBGEN program, the remote site, or both.

Figure 2 shows the processes at the remote site. After the operating system and libraries are restored, the IMS and Db2 activities can run in parallel as long as enough resources are available. CR DR PiT TS is the coordinated recovery point.

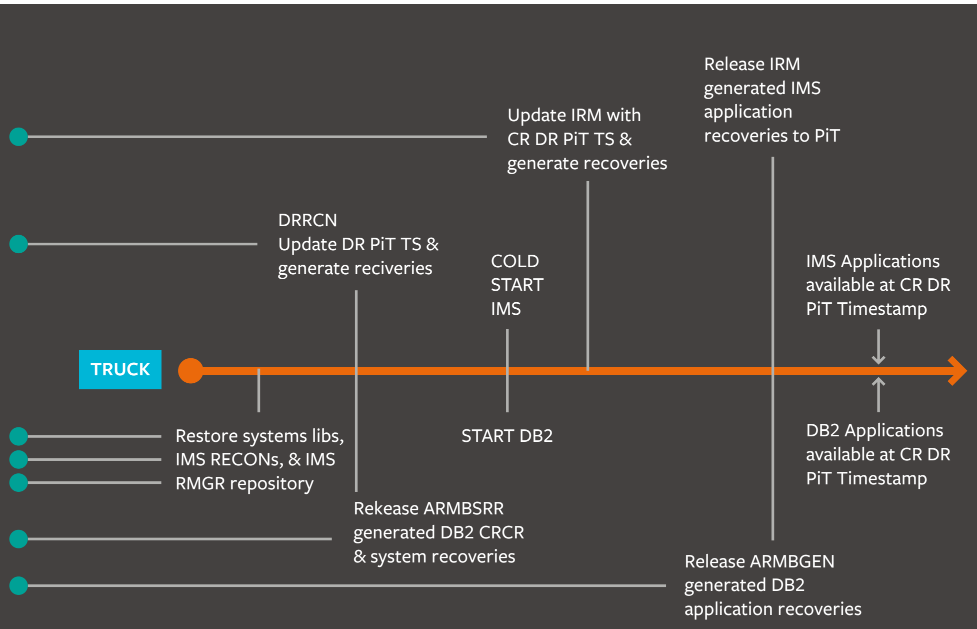

Release IRM generated IMS application recoveries to PiT
Update IRM with CR DR PiT TS & generate recoveries
DRRCN Update DR PiT TS & generate reciveries
COLD START IMS
IMS Applications available at CR DR PiT Timestamp
TRUCK
Restore systems libs, IMS RECONs, & IMS RMGR repository
START DB2
DB2 Applications available at CR DR PiT Timestamp
Rekease ARMBSRR generated DB2 CRCR & system recoveries
Release ARMBGEN generated DB2 application recoveries

Figure 2 - Coordinated Recovery Process at the Remote Site

THE REXX EXECS

The following REXX programs and process descriptions are delivered as is, and no warranties, expressed or implied, are provided with these shareware solutions. Use them at your own risk, and feel free to modify them as you see fit. Refer to the BMC Software product documentation sets for the most complete details about product usage and other tips. For example, the RECOVERY MANAGER for IMS User Guide provides assistance in planning and implementing the IMS portion of a DR process.

The IMS and Db2 REXX EXEC—CRPREXX

A REXX program that captures the IMS recovery point timestamp and passes it to the ARMBTSI program follows. The high-level process flow is as follows:

1. Browse the output of the RMGR for IMS DRRCN utility.
2. Obtain the timestamp value from the LATEST PIT RECOVERY TIME IS: value field.
3. Convert the timestamp.
4. Present the converted timestamp to the RMGR for Db2 ARMBTSI program, which inserts the value into a repository table (CRRDRPT).

As written, the DRRCN report is expected to be in 'BMCIRM.BB.DRRCN.REPORT' and the local offset from GMT is expected to be in 'BMCIRM.BB.DRRCN.OFFSET'. The offset should be in the first three columns: +05, -11, and so on. The JCL for ARMBTSI is in 'BMCIRM.BB.DRRCN.ARMBTSI.CNTL' and the REXX routine updates the TIMESTAMP in the PARM string in place.

Note: The offset file 'BMCIRM.BB.DRRCN.OFFSET' is used only if option TIMEVALUE=GMT is specified (RMGR for IMS DRRCN utility). The default is to show all times in local format in which case the timestamp does not have to be adjusted.

You can modify the REXX as appropriate to your shop standards.

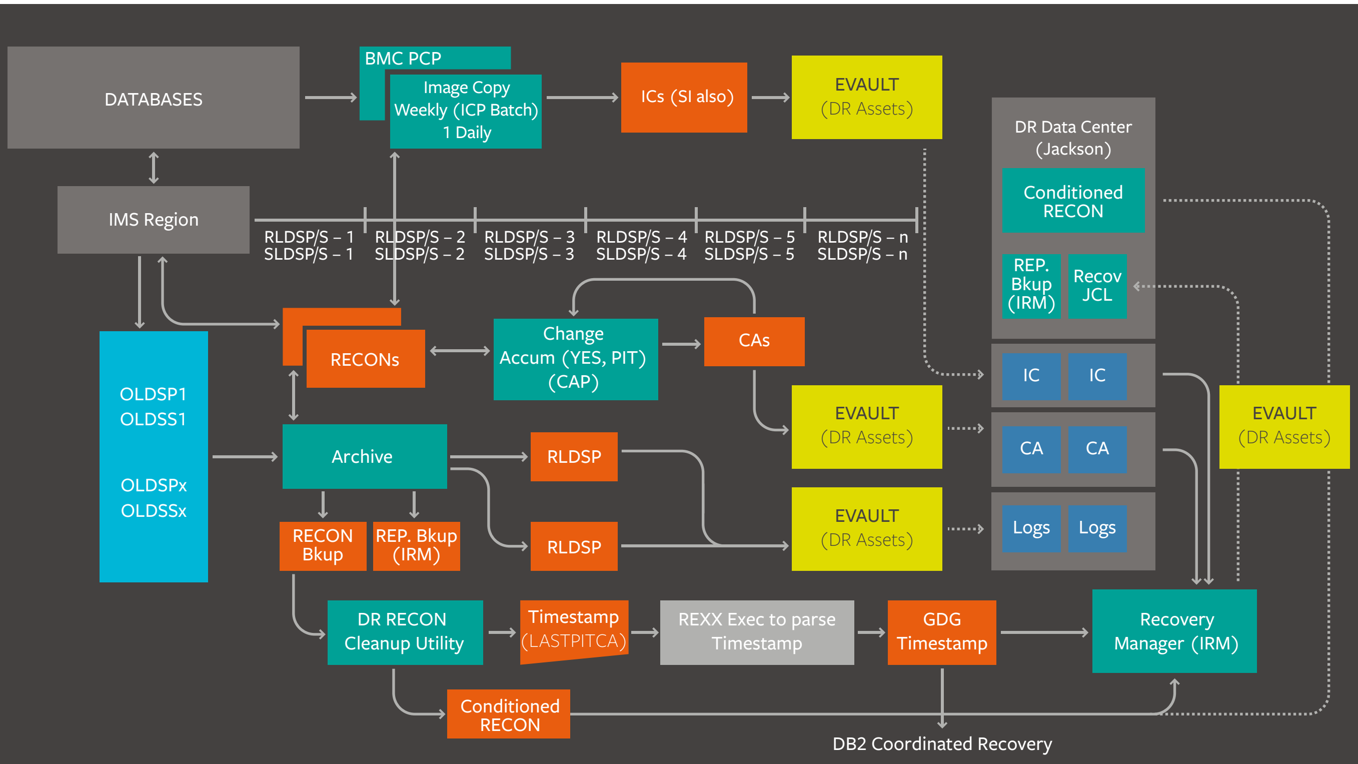
Note: The REXX EXEC CRPREXX examines the output of the RMGR for IMS DRRCN utility and is therefore dependent on the format and content of that output. Any implementation of this REXX EXEC should be re-examined, modified accordingly, and tested when any new release of the DRRCN utility is installed (the DRRCN utility is provided with RECOVERY MANAGER for IMS and the Backup and Recovery Solution for IMS product).

Here’s the REXX EXEC for IMS and Db2:

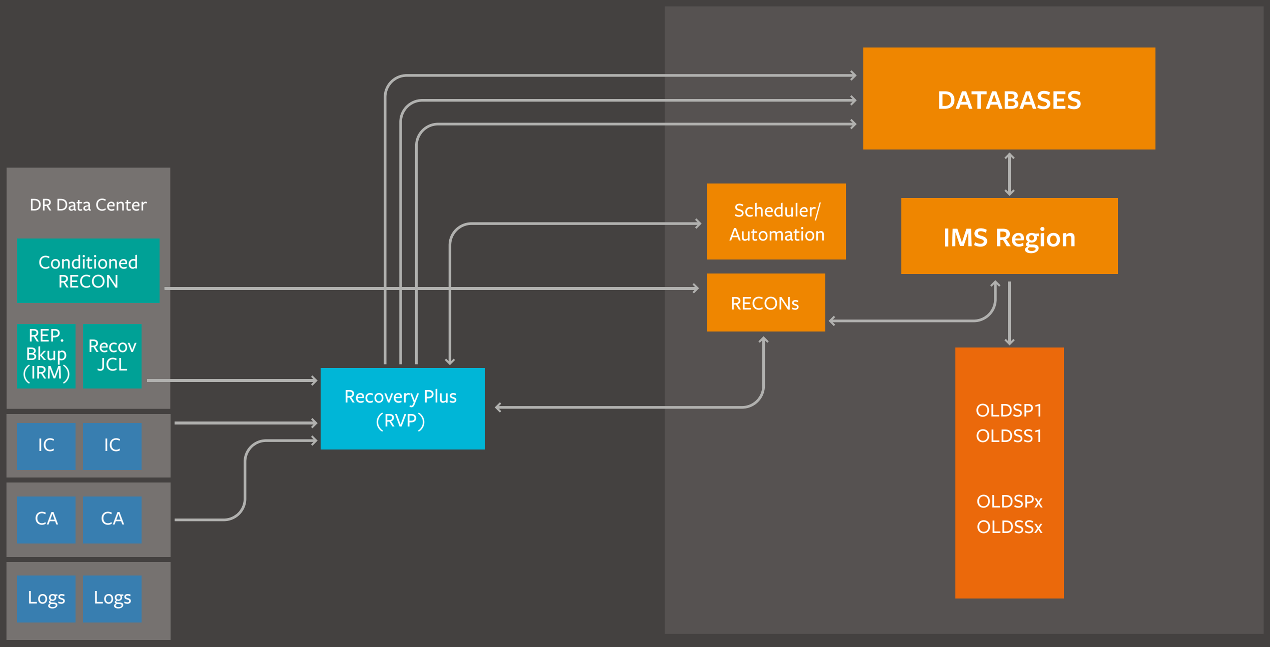
```
/* REXX */
/******************************************************************/
/*   GET RECOMMENDED PIT TIMESTAMP FROM DRRCN REPORT              */
/******************************************************************/
ADDRESS TSO
 "ALLOC DD(INDD) DSNAME('BMCIRM.BB.DRRCN.REPORT')  SHR REUSE"
 IF RC ^= 0 THEN
   DO
    SAY 'UNABLE TO ALLOCATE DRRCN REPORT - RC = 'RC EXIT 12
   END
 local = 'NO'
 eof = 'NO'
 DO WHILE eof = 'NO'
   "EXECIO 1 DISKR INDD"
    IF RC = 0 THEN
      DO
       PULL record
       IF INDEX(record,'ALL TIMES ARE IN LOCAL') > 0 THEN local = 'YES'
       IF INDEX(record,'LATEST PIT RECOVERY TIME IS') > 0 THEN
         DO
           IF INDEX(record,'(N/A)')  > 0
             THEN DO
               SAY 'LATEST PIT RECOVERY TIME IS UNAVAILABLE'
               EXIT 8
             END
           yy    = SUBSTR(record,34,2)
           ddd   = SUBSTR(record,36,3)
           hour  = SUBSTR(record,40,2)
           min   = SUBSTR(record,43,2)
           sec   = SUBSTR(record,46,2)
           tenth = SUBSTR(record,49,1)
           imsdate = yy||ddd||' '||hour||':'||min||':'||sec||'.'tenth
           eof = 'YES'
         END
      END
    ELSE eof = 'YES'
 END
 "EXECIO 0 DISKR INDD (FINIS"
 "FREE DD(INDD)"
SAY 'IMS  GMT  TIMESTAMP IS: ' imsdate
IF local = 'NO' THEN
  DO
    /****************************************************************/
    /*   CHECK AND ADJUST FOR LEAP YEAR                             */
    /****************************************************************/
     yeardays = 365 pyeardays = 365 chkdate = 2000 +yy
     IF TRUNC(chkdate/4)  = chkdate/4 THEN yeardays = yeardays + 1
     IF TRUNC((chkdate-1)/4)  = (chkdate-1)/4 THEN
      pyeardays = pyeardays + 1
  /****************************************************************/
  /* GET OFFSET & ADJUST GMT TIMESTAMP TO LOCAL TIME              */
  /****************************************************************
   / "ALLOC DD(IND2) DSNAME('BMCIRM.BB.DRRCN.OFFSET')  SHR REUSE"
   IF RC ^= 0 THEN
     DO
       SAY 'UNABLE TO ALLOCATE OFFSET DATA SET - RC = 'RC
       EXIT 12
     END
   eof = 'NO'
   DO WHILE eof = 'NO' "EXECIO 1 DISKR IND2"
      IF RC = 0 THEN
        DO
         PULL record
          offset = SUBSTR(record,2,2)
          IF SUBSTR(record,1,1)  = '-' THEN
            DO
              hour = hour - offset
              IF hour < 0 THEN
                DO
                  hour = hour + 24
                  ddd = ddd - 1
                END
              IF ddd = 0 THEN
                DO
                  ddd = pyeardays
                  yy  = yy - 1
                END
            END
          ELSE
            DO
              hour= hour + offset
              IF hour >= 24 THEN
                DO
                  hour = hour - 24
                  ddd = ddd + 1
                END
              IF ddd > yeardays THEN
                DO
                  ddd = 1
                  yy =  yy + 1
                END
            END
        END
      ELSE eof = 'YES'
   END
   "EXECIO 0 DISKR IND2 (FINIS"
   "FREE DD(IND2)"
END
```

```
IF local = 'YES' THEN SAY 'PIT RECOVERY TIME IS LOCAL'
ELSE SAY 'OFFSET FROM GM TIME IS:  'SUBSTR(record,1,1)||offset
/****************************************************************/
/*   BUILD DB2 TIMESTAMP                                          */
/******************************************************************/
 yyddd = RIGHT(JUSTIFY(yy,LENGTH(yy)),2,'0')|,
         RIGHT(JUSTIFY(ddd,LENGTH(ddd)),3,'0')
 hour  = RIGHT(JUSTIFY(hour,LENGTH(hour)),2,'0')
 tmpdate =DATE('S',yyddd,'J')
 year    = SUBSTR(tmpdate,1,4)
 month   = SUBSTR(tmpdate,5,2)
 day     = SUBSTR(tmpdate,7,2)
 db2date = year||'-'||month||'-'||day||'-'||hour||'.'||min||'.'||sec
SAY 'DB2 LOCAL TIMESTAMP IS: ' db2date
/******************************************************************/
/*   INSERT THE DB2 TIMESTAMP                                       */
/******************************************************************/ "ALLOC DD(IND4)
 DSNAME('BMCIRM.BB.DRRCN.ARMBTSI.CNTL')  OLD REUSE"
 IF RC ^= 0 THEN
   DO
     SAY 'UNABLE TO ALLOCATE ARMBTSI DATA SET - RC = 'RC
     EXIT 12
   END
 eof = 'NO'
 foundit = '0'
 searchstring = 'PGM=ARMBTSI'
 DO WHILE (eof = 'NO' & foundit = '0')
   "EXECIO 1 DISKRU IND4"
    IF RC = 0 THEN PULL record
    ELSE eof = 'YES'
    IF INDEX(record,searchstring)  > 0 THEN
      DO
        IF searchstring = 'PARM=' THEN foundit = '1'
        ELSE searchstring = 'PARM='
        IF INDEX(record,searchstring)  > 0 THEN foundit = '1'
      END
 END
  IF foundit = '1' THEN
    DO
      record = OVERLAY(db2date,record,INDEX(record,'PARM=')+11)
      QUEUE record
      "EXECIO 1 DISKW IND4"
    END
  ELSE SAY 'UNABLE TO INSERT ARMBTSI TIMESTAMP'
  "EXECIO 0 DISKR IND4 (FINIS"
  "FREE DD(IND4)"
```

COORDINATED IMS AND DB2 LOCAL AND REMOTE SITE EXAMPLE DIAGRAM



Current Backup and Recovery Diagram



IMS Disaster Recovery Site Procedures

FOR MORE INFORMATION

To learn more about BMC Coordinated Recovery, please visit **bmc.com/lorem_ipsum**

BMC is a global leader in innovative software solutions that enable businesses to transform into digital enterprises for the ultimate competitive advantage. Our Digital Enterprise Management solutions are designed to fast track digital business from mainframe to mobile to cloud and beyond.

BMC digital IT transforms 82 percent of the Fortune 500.

496548