Best Practices for DB2 on z/OS Backup and Recovery

Susan Lawson and Dan Luksetich
www.db2expert.com
and
BMC Software
June 2009
Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC</td>
<td>713 918 8800 or 800 841 2031</td>
<td>713 918 8000</td>
</tr>
<tr>
<td>2101 CITYWEST BLVD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Outside United States and Canada

<table>
<thead>
<tr>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>(01) 713 918 8800</td>
<td>(01) 713 918 8000</td>
</tr>
</tbody>
</table>

© Copyright 2009 BMC Software, Inc.

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners.

DB2 and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

BMC Software considers information included in this documentation to be proprietary and confidential. Your use of this information is subject to the terms and conditions of the applicable End User License Agreement for the product and the proprietary and restricted rights notices included in this documentation.

YL&A is an internationally recognized firm specializing in DB2 database and application design, performance and availability. They have provided solutions for some of the worlds largest and most complex databases. They are also authors of numerous industry articles and books including the 'IBM DB2 9 z/OS DBA Certification Guide'. To contact them visit www.db2expert.com.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.

Customer support

You can obtain technical support by using the BMC Software Customer Support website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, see “Before contacting BMC.”
# Contents

## Chapter 1  Introduction to Backup and Recovery  
7

- Business continuity .................................................................................................................. 8
- Recovery objectives ..................................................................................................................... 9
- Backup and recovery components. ............................................................................................... 9
  - Image copies ................................................................................................................................. 10
  - Bootstrap data set (BSDS) ........................................................................................................ 10
  - Log ........................................................................................................................................... 11
  - Catalog and directory ................................................................................................................. 12
- YL&A backup and recovery recommendations ........................................................................... 12
  - Take frequent copies ................................................................................................................ 12
  - Take and keep a copy on DASD ................................................................................................. 13
  - Take partition-level copies ....................................................................................................... 13
  - Use parallelism .......................................................................................................................... 13
  - Recover multiple objects in a list ............................................................................................. 13
  - Put recent archived logs on DASD ........................................................................................... 13
  - Optimize recovery with system installation parameters (ZPARMs) ..................................... 14
  - Automate your tasks ................................................................................................................. 14

## Chapter 2  Backup and recovery utilities  
15

- Image copy ................................................................................................................................ 16
  - COPY-Pending status .............................................................................................................. 17
- Image copy types ....................................................................................................................... 18
- Index image copies ..................................................................................................................... 19
- SHRLEVEL impact ..................................................................................................................... 20
- Concurrent copy ......................................................................................................................... 20
- Copy parallelism ........................................................................................................................ 20
- LISTDEF .................................................................................................................................... 21
- Stacked data sets ...................................................................................................................... 22
- Quiesce ....................................................................................................................................... 22
- Recover ....................................................................................................................................... 23
  - Recover to current ...................................................................................................................... 24
  - Recover to a point in time ......................................................................................................... 24
  - Recover objects concurrently .................................................................................................. 26
  - Copy impacts on recovery ....................................................................................................... 26
  - Recover the DB2 catalog and directory ................................................................................... 27
  - Disaster recovery ..................................................................................................................... 28
- Backup and restore system capability. .......................................................................................... 28
  - Object-level recoveries ........................................................................................................... 29
- BMC backup and recovery utilities ............................................................................................ 30

## Chapter 3  Advanced Backup and Recovery  
31

- Overview .................................................................................................................................... 32
- Snapshot/FlashCopy .................................................................................................................. 32
  - Snapshot technology ............................................................................................................... 33
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Disaster Recovery Planning</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>Overview</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>The disaster database</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>Disaster recovery health queries</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>Generating jobs</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>BMC techniques for disaster recovery preparation</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Disaster and local site recovery simulation</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>Disaster recovery estimation</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>Disaster recovery evaluation</td>
<td>48</td>
</tr>
<tr>
<td>5</td>
<td>Data replication</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>Data replication techniques</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Unload and load</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Log analysis and replication</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>DSN1COPY</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>Snapshot/FlashCopy data set level copies</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>Data replication using BMC technologies</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>Standard DBA Techniques</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>Creative DBA Techniques</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>Choose the Right Technique</td>
<td>54</td>
</tr>
<tr>
<td>6</td>
<td>BMC best practices for business continuity and availability</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>Overview</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>Objective of advanced technology</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>BMC Recovery Management for DB2</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>Smart copy techniques</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>Hybrid copy techniques</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>Snapshot copies</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>Cabinet Copy</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>Data set level instant snapshot</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>Incremental index copy</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>A major issue: consistent image copy</td>
<td>71</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>BMC Software smart recovery techniques</td>
<td>75</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------------------</td>
<td>----</td>
</tr>
<tr>
<td>Overview</td>
<td>........................................</td>
<td>76</td>
</tr>
<tr>
<td>Point-in-time recovery</td>
<td>..................................</td>
<td>76</td>
</tr>
<tr>
<td>Backward-to-forward recovery automation</td>
<td>.......................</td>
<td>78</td>
</tr>
<tr>
<td>In-flight resolution technology and timestamp recovery</td>
<td>.................</td>
<td>79</td>
</tr>
<tr>
<td>Recovery avoidance</td>
<td>..................................</td>
<td>81</td>
</tr>
<tr>
<td>Object recovery avoidance</td>
<td>................................</td>
<td>81</td>
</tr>
<tr>
<td>Conditional restart avoidance</td>
<td>................................</td>
<td>82</td>
</tr>
<tr>
<td>Data-set-level instant restore</td>
<td>.........................</td>
<td>83</td>
</tr>
<tr>
<td>Automatic index recovery to rebuild</td>
<td>..........................</td>
<td>84</td>
</tr>
<tr>
<td>Dropped object recovery automation</td>
<td>.........................</td>
<td>85</td>
</tr>
<tr>
<td>Transaction recovery using undo and redo SQL</td>
<td>..................</td>
<td>85</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 8</th>
<th>The right solution for every backup and recovery need</th>
<th>87</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC Recovery Management for DB2 solution</td>
<td>......................</td>
<td>88</td>
</tr>
<tr>
<td>Which products/solutions are right for me?</td>
<td>................................</td>
<td>90</td>
</tr>
<tr>
<td>Frequently asked questions</td>
<td>................................</td>
<td>91</td>
</tr>
</tbody>
</table>
As the size and complexity of our databases continue to grow at a tremendous pace, the need for an appropriate backup and recovery strategy becomes more critical. At the same time, the choices for backup and recovery options are becoming more complex. Who performs the backup and recovery? Is it the DBA, the system programmer, or the DASD administrator? The choices made, and the personnel involved, are critical for the proper backup and recovery strategy. With more and more databases being placed into a single subsystem or data sharing group, can each one have the same backup and recovery strategy? If not, how does the strategy for a single database affect the other databases? How does the local backup and recovery plan compare to the disaster recovery plan, and what is the recoverability of your databases right now? These are complex questions that require a complete local and remote backup and recovery plan that is well coordinated between DASD personnel, system programmers, DBAs, application developers, management, and users.

This best practices guide covers not only the mechanics of DB2 backup and recovery, but also the impact of various options upon the applications using the databases. This guide addresses various simple and advanced backup and recovery strategies, and covers offerings from DB2 itself, from DASD vendors, and from BMC Software. Image copy and recovery utilities are discussed, along with advanced strategies involving DASD mirroring, log replication, and multi-site operations. When a disaster hits will there be all the documentation needed to recover the databases? This guide addresses how to automate a disaster recovery plan such that only the most minimal of documentation is actually needed at the hot site. The guide includes extras such as non-standard backup and recovery methods, as well as fast replication techniques for making quick copies of data.

Susan Lawson and Daniel Luksetich of YL&A wrote much of the general material, and BMC supplied the product-specific text. Susan Lawson and Dan Luksetich are internationally recognized DB2 consultants. They specialize in DB2 performance and availability. They have both authored several articles and books on DB2. For more information or to contact them visit www.db2expert.com.

For more information on any BMC products, refer to the product documentation or www.bmc.com/db2.
Chapter 1 Introduction to Backup and Recovery

By Susan Lawson and Daniel Luksetich, YL&A

Business continuity ................................................................. 8
Recovery objectives ............................................................... 9
Backup and recovery components ......................................... 9
  Image copies ................................................................. 10
  Bootstrap data set (BSDS) .................................................. 10
  Log ................................................................................. 11
  Catalog and directory ....................................................... 12
YL&A backup and recovery recommendations ............................. 12
  Take frequent copies ........................................................ 12
  Take and keep a copy on DASD .......................................... 13
  Take partition-level copies ................................................. 13
  Use parallelism .................................................................. 13
  Recover multiple objects in a list ........................................ 13
  Put recent archived logs on DASD ....................................... 13
  Optimize recovery with system installation parameters (ZPARMs) 14
  Automate your tasks ......................................................... 14
Business continuity

A business continuity plan provides for an enterprise level plan to counteract interruptions of business activities, and to protect critical business processes from failures or disasters—in other words, to keep you from going out of business in the event of a failure or disaster. The plan relates to all business processes and needs to include all methodologies and procedures to keep the business operating at acceptable levels. A strategy for backup and recovery should be part of your business continuity plan.

The backup and recovery plan is the IT subset of the business continuity plan and relates only to the information systems (computing hardware, software, data copies and recovery procedures). You need plans for both local and remote backup and recovery.

Prepare for various types of disasters and failures, including local and site disasters. Each has special considerations when planning a strategy. In general, a disaster is an unplanned event that results in the loss of operation of information processing activities. This can result from damage to hardware maybe from fire, flood, malice, or machine failure. It could also be the result of damage to software/data which could also have resulted from machine failure, malice or program data corruption.

- A local disaster is where the computer center is operational, however one or more components are damaged or inoperable.

- A site disaster is where the entire site is destroyed or damage beyond functioning capabilities, or it could also result from a protracted local disaster.

Creating proper backup and recovery procedures is a critical part of operating a desired 24x7 environment. There are many technologies to understand and utilize. Some of technologies are included in the DB2 product, but it is always necessary to supplement this with utilities and tools. You always need to take image copies. You should have a contingency plan if your primary plan fails, and that contingency plan should rely on DB2 image copies and DB2 recovery.

Your decision on an overall strategy must start with a cost/benefit analysis. You need to obtain realistic recovery point objectives (RPOs) and recovery time objectives (RTOs) that can be balanced with a budget. You need to determine an application recovery sequence. Every application is not the most important application, and every application does not have to be recovered first. Anarchy is not a useful recovery plan! Management needs to clearly understand these relationships, agree on application priorities, and be realistic about the money they want to spend.
Recovery objectives

When creating backup and recovery plans, ask yourself what your objectives are. You may want all the data available all the time, but this is often not realistic due to two things: time and money. Establish your objectives first, and then plan accordingly:

- **RTO** - the amount of time within which a business process must be restored after an unplanned outage or disaster
- **RPO** - the point in time to which the database is to be restored after an unplanned outage or disaster

Different RTOs and RPOs may apply for local disasters and for site disasters. There is a direct service-level-to-cost relationship between RTO and RPO expectations. Be sure to balance management expectations with reality. To say that you want all the data to be completely available all the time is *not* a backup and recovery strategy, but rather a need for high availability. They are related but also different. If you need to restore data immediately after a failure with no data loss, be prepared to pay. In many cases, you must decide which critical data and applications need to be recovered as quickly as possible; recovering everything immediately may be impossible and usually is not necessary to keep the critical workloads running.

RTOs and RPOs are directly related to cost: as the RTO decreases, the costs increase; as the RPO increases, the costs also increase.

To establish a backup and recovery plan, determine your business continuity objectives for all applications and for critical applications. You can set application-level priorities or system-wide priorities. Consider aspects such as legal requirements and costs of losing business and customers due to data loss and/or downtime. Measure downtime in days or hours. Of course, consider expectations from management—they must be realistic. Management must understand the costs:

- The lower the RTO the higher the cost.
- The greater the RPO the higher the cost.

Backup and recovery components

One of the great advantages of database management systems is that backup and recovery is built into the database software. DB2 supports and documents a variety of backup and recovery strategies through the DB2 log, bootstrap data sets, system catalog and directory, copies, utilities, and operating system components. With this set of tools, DB2 enables recovery to current state (up to point of failure) or to an earlier state.
DB2 supports the following units of recovery:

- table spaces
- indexes
- index spaces
- partitions
- data sets
- a page or range of pages
- the entire system

### Image copies

To be able to recover data when a problem occurs, you need a solid recovery plan and a strategy for each of the many situations that can arise. A key part of your plan should include taking image copies on a regular basis. You can create image copies of your table spaces and index spaces. How often image copies should be taken depends on the allowable down time for the primary user of an application. Determine this clock-time requirement ahead of time, in case you have to recover your data. For example, if an application can’t be unavailable for more than two hours, you’ll need to take image copies more frequently than if the application can be unavailable for a complete day. Depending on the currency of data required in the recovery, determine when image copies can be taken and what types of image copies to take. You may need to take incremental image copies throughout the day, or you may need to do only one full copy at the end of the day’s processing. In some situations, especially where long-running batch processes are involved, a best practice is to take an image copy at the beginning of a batch window; this enables a restore in case something goes wrong during the batch cycle.

### Bootstrap data set (BSDS)

A key component to DB2 recovery is the bootstrap data set (BSDS). The BSDS is a VSAM key-sequenced data set that contains information about the DB2 logs, the records contained in those logs, and other information. DB2 records all the current active log data sets in the BSDS. During offload processing, DB2 dynamically allocates a new archived log data set with a unique name. After a successful offload, this data set is recorded in the BSDS, and DB2 creates a copy of the BSDS. When the archive log is placed on a cartridge, the BSDS copy is placed on the same cartridge. During a recovery, DB2 uses the BSDS to find all the available archive logs.
The DSNZPARM parameter MAXARCH determines the number of records the BSDS can contain. The BSDS should be large enough to record all archive logs (maximum 10000). The proper setting for MAXARCH depends on how large your archive logs are and on the oldest point in time to which an application is allowed to recover. Be sure to set the MAXARCH parameter high enough for your environment, and retain your archive log data sets long enough so that you’ll have them if a recovery requires you to go back through older archive logs.

---

DB2 writes one or more records to its log in order to be able to back out changes if the unit of work fails when changes are made to the data in tables, causing a subsequent change to the table space and index spaces. DB2 can also use the log information to reapply changes that may have been lost when recovering to a previous point in time. The primary purpose of the active log is to record all changes to the DB2 object data (inserts, updates, and deletes). DB2 records the Data Manipulation Language (DML) statements in the log as follows:

- During an insert, the entire after-image of the record is logged (redo record).
- During a delete, the before-image is recorded (undo record).
- During an update, both the before and after images are recorded (undo and redo records).

Each log record has its own unique identifier that represents the offset of the record in the log from the beginning of the log:

- In a non–data sharing environment, the identifier is known as the log relative byte address (RBA).
- In a data sharing environment, this identifier is known as the log record sequence number (LRSN). The LRSN helps track the sequence of events that happen over multiple members (subsystems) in the data sharing group.

DB2 physically records changes in log data sets. Each DB2 subsystem has a predefined set of active logs on disk to which the log records are written. When the active log data sets become full, DB2 automatically switches to the next available log data set. When all the active log data sets have been used, DB2 wraps around to the first active log. The active log data sets are offloaded when they become full. The offloaded active logs are known as archive log data sets and can be stored on disk or on cartridge tapes. There can be multiple copies of archive log data sets.

Recovery from active logs is faster than recovery from archive logs, and recovery from disk archive log data sets is much faster than from tape archive logs because you can take advantage of DB2’s ability to process the logs in parallel.
You must keep several archive log data sets in case a recovery is necessary. The retention period of the archive log data sets depends on the image copy frequency and how far back in time you go to do a point-in-time recovery. Up to 10,000 archive logs can be kept.

Dual-logging capability ensures that two copies of the active log data sets exist; if one is lost, the other can be used for recovery. During a recovery, DB2 applies the changes recorded in the active log that are required to recover to the specified point in time. If the records required for recovery are no longer on the active log, DB2 calls for the appropriate archive logs. All production systems should be using dual logs to ensure that data can be recovered successfully. As a best practice, size the active logs to keep 24 to 48 hours of log data.

Catalog and directory

The DB2 catalog and directory are key components to performing backup and recovery. If you lose your catalog and/or directory, you will not have an operational DB2 subsystem. The catalog and directory both contain information critical for recovery. For example, the catalog table SYSIBM.SYSLGRNX in the DB2 directory gives DB2 the ability to determine which log records are needed to recover a table space and index. The SYSLGRNX table contains a row for each time period during which update activity occurred for a table space or index space. The catalog contains important information that can be referenced when building and executing recovery processes for your objects.

During a system-wide recovery, DB2 recovers the catalog and directory first; no other recoveries are possible until this process is complete. Copy the catalog and directory often to make the recovery process for these objects as fast as possible. The time required to recover the catalog and directory is crucial and must be minimized.

YL&A backup and recovery recommendations

Here are some tips to improve the CPU and elapsed times of your backup and recovery jobs, as well as make your job a little easier and your data a little safer.

Take frequent copies

Take copies often to reduce the amount of log data required for a recovery. At the very least, take meaningful copies that reflect the activity of the objects being copied. Make sure you take primary and backup copies, and then copy the copies for offsite shipment.
Take and keep a copy on DASD

Image copies run much faster when written to DASD. DFSMShsm or a manual process can move them to tape after the fact, and copy them for offsite storage. However, most people focus on the backup and not the recovery. Recovery runs much faster from DASD than from tape, and the issues with tape stacking do not come into play when you recover from DASD. You can get the highest level of backup and recovery parallelism if you put your copies on DASD.

Take partition-level copies

A partition-level copy can support both a partition-level recovery and full recovery one partition at a time. Recovery of a single partition can be faster from a partition level copy. You can employ a higher level of parallelism with partition-level copies.

Use parallelism

This can be either utility controlled parallelism or manual parallelism. Manual parallelism is better but much more work. A nice balance between the two is best.

Recover multiple objects in a list

This eliminates multiple passes through the log, and can also improve the processing of copies on tape if the copy data sets are stacked.

Put recent archived logs on DASD

You can use either a manual process or use DFSMShsm to move the archived logs from DASD to tape as they age. DASD archived logs get you faster application of log records and a higher degree of parallelism for your recoveries.

Use larger quantities of active logs. Application of log data in a recovery from the active logs is fastest!
Optimize recovery with system installation parameters (ZPARMs)

- DLDFREQ – frequency of LEVELID and HPGBRBA updates
- LOGAPSTG – Fast log apply storage
- CHECKFREQ – Checkpoint frequency
- PCLOSEN – RO switch checkpoints
- PCLOSET – RO switch time

Automate your tasks

You can use SQL to read the SYSCOPY catalog table to get information about the status of an object (its recoverability) and to generate backup and recovery jobs.
Chapter 2 Backup and recovery utilities

This chapter discusses the main backup and recovery utilities: image copy, quiesce, and recover.

By Susan Lawson and Daniel Luksetich, YL&A

Image copy ......................................................... 16
COPY-Pending status .................................................. 17
Image copy types ...................................................... 18
Index image copies .................................................... 19
SHRLEVEL impact ..................................................... 20
Concurrent copy ....................................................... 20
Copy parallelism ....................................................... 20
LISTDEF ............................................................. 21
Stacked data sets ...................................................... 22

Quiesce ............................................................... 22

Recover ............................................................... 23
Recover to current ..................................................... 24
Recover to a point in time ........................................... 24
Recover objects concurrently ..................................... 26
Copy impacts on recovery ......................................... 26
Recover the DB2 catalog and directory ......................... 27
Disaster recovery .................................................... 28
Backup and restore system capability .............................. 28
Object-level recoveries ............................................. 29

By BMC Software

BMC backup and recovery utilities ................................ 30
Image copy

The DB2 image copy is the primary and most basic way to back up DB2 data. The COPY utility, which is used to take image copies, can be invoked directly or as part of another utility execution (typically REORG or LOAD). There are generally two types of image copies: full or incremental. Depending upon the utility software you are using, you can create several copies of a DB2 object, record the information about the copies in the DB2 system catalog, and report on the change status of the table space. Copy utilities typically reset the COPY-Pending status or informational COPY-Pending status of a DB2 object.

When image copies are needed depends upon a variety of conditions and requirements. You have lots of flexibility with the quantity, type, and concurrency of image copies; therefore, a number of questions should be answered to determine the copy need:

- What are the recovery requirements?
- How long are logs retained?
- How much storage is available for backups?
- How often are objects updated?
- What is the RTO?
- How is an object related to other objects?

Remember that your data is in a relational database. Objects that are related to each other may need to be backed up together. Some objects are static, and others are updated quite often. Objects are sharing storage, and applications are sharing objects. These factors come into play when determining backup frequency. Your requirements will determine the copy type and the frequency of the copies:

- Number of copies to be taken
- The type of copy taken
  - Incremental or full
  - Local, backup, remote
- Copy concurrency
- The copy media used
- How is the copy is actually taken
  - Copy software
  - Copy hardware

Image copies are typically required after a REORG or LOAD operation or after a mass update of the data in a table. A REORG or LOAD utility generally adds or reorganizes large quantities of data, and usually does not log the changes to the data. Copying the objects after these operations is usually required by DB2, and is strongly recommended.
Image copies are not always required, but are strongly recommended both before and after an ALTER to an object. The most conservative approach to altering an object would be to take an image copy, alter the object, REORG the object, and then take another image copy. DB2 allows online schema changes. It creates and maintains “versions” to maintain availability, at the price of performance and some risk to backup and recovery. Because REORGs and image copies can be taken online, you can have a conservative approach to altering an object while it remains available. By taking this conservative approach to altering objects, information is recorded in the SYSIBM.SYSCOPY catalog table so that you know the start and stop points of the changes (via the COPY and REORG), and the object is recoverable to the points before and just after the ALTER.

One final thought on the need for image copies: you always need image copies! The image copy is the built-in backup and recovery mechanism for DB2. Even if you have advanced recovery options (hardware) in place, those recovery options may fail. If they do, the contingency would be a DB2 RECOVER—which requires an image copy. DB2 is always recording log ranges in SYSLGRNGX in the directory. The MODIFY utility is used to prune the log range information as it ages, but the MODIFY utility can run only if a copy exists. Therefore, if you want to maintain a healthy directory that is not over-sized, you must take image copies.

**COPY-Pending status**

DB2 must ensure that the data is recoverable. Certain events, such as unlogged utilities, can force a table space or table space partition into COPY-Pending status. An object in COPY-Pending status is not recoverable to a current state and perhaps not recoverable at all. An object in COPY-Pending status is also not updateable.

A full image copy is the most reliable way to get an object out of COPY-Pending status. You can also get an object out of COPY-Pending status by using these methods:

- LOAD REPLACE LOG YES or LOAD REPLACE with an inline image copy
- RECOVER to a prior recoverable point in time
- Remove the COPY-Pending status (*not recommended*)
  - START command with FORCE option
  - REPAIR utility with SET NOCOPYPEND
  - LOAD REPLACE with LOG NO NOCOPYPEND option

An additional status is ICOPY-Pending status. The “I” stands for informational, and it applies to both indexes and table spaces. The ICOPY-Pending status can be set if an index is created with the COPY YES option, or a table space is created with the LOG NO (or NOT LOGGED) option. An object in ICOPY-Pending status is not recoverable to current or perhaps not recoverable at all.
Certain events will place an object in ICOPY-Pending status. Indexes that have the COPY YES option set will be placed in ICOPY-Pending when there is a LOAD REPLACE LOG NO or REORG LOG NO utility executed. Table spaces defined with LOG NO (or NOT LOGGED) are set in ICOPY-Pending status when any change is made to the table space.

You can take objects out of ICOPY-Pending by using the following methods:

- Full image copy of the object
- LOAD REPLACE of a table space with an inline copy
- RECOVER to a recoverable prior point in time
- ALTER INDEX COPY NO
- Remove the ICOPY-Pending status (*not* recommended)
  - START command with option FORCE
  - REPAIR utility with SET NOCOPYPEND
  - LOAD REPLACE with LOG NO NOCOPYPEND option

### Image copy types

Image copies can be full or incremental.

A full image copy creates a full copy of the object or objects specified, including:

- Table space
- Index space
- Partition (table space or index space)

Depending upon the utility software you are using, full image copies are the default.

Incremental image copies create a copy of only the changes since the previous image copy. Incremental copies can provide a performance and space savings advantage over full image copies because a complete copy is not taken; only the changed pages are copied. This obviously saves storage for backups, but may also reduce the amount of elapsed and CPU time it takes to create the copy.

An incremental copy uses the space map pages in the page set to determine which pages have changed and thus need to be backed up. The TRACKMOD table space setting will affect the performance of the image copy. If TRACKMOD is set to NO, the space map pages will not indicate which pages to copy and the incremental copy must scan the entire table space. Therefore, performance may not be better than a full copy. The TRACKMOD setting is used to improve heavy update performance in a
data sharing environment where P-locking on the space map can become an issue. Our tests have not demonstrated this particular performance advantage, but your results may vary. Just remember the trade-offs when it comes to the incremental image copy capability.

Incremental image copies are not always allowed. They are never allowed for indexes, and they are only allowed for a table space when the following is true:

- A full image copy exists.
- The COPY-Pending status is not on.
- The last copy taken was not a concurrent copy.
- This is not the first copy after a CREATE.
- This is not the first copy taken after a LOAD or REORG.
- The previous COPY utility was not cancelled.

Your backup software may have fewer or more requirements. Typically, when an incremental is not allowed a full copy will automatically be taken.

**Index image copies**

To take image copies of index spaces and index space partitions, specify COPY YES on the CREATE INDEX statement. The default is COPY NO. To take an image copy of an index space, be sure that the COPY YES option is set.

You may want to enable image copies of large indexes because recovering the index may be faster than rebuilding it. In addition, a recovery of an index can be performed concurrently with the recovery of its associated table space; a rebuilding of an index cannot happen until the associated table space is recovered. For very large indexes, there may be constraints in the CPU and temporary storage for rebuilding. In these cases, copying the index may be the only way to guarantee recoverability.

Compressed indexes are copied uncompressed, so consider this when estimating backup storage requirements. This is an important point because this is different from table space compression, where the compressed data is copied.
SHRLEVEL impact

The SHRLEVEL option of the image copy affects the concurrency of other processes accessing the objects being copied. The option you select not only affects the concurrency of the copy process, but also the stability of the copy and the recoverability of the object being copied:

- SHRLEVEL REFERENCE copies only allow readers while the copy is executing. A SHRLEVEL REFERENCE image copy is a stable copy, and consistency is guaranteed if you recover to that copy. If a list of objects is copied SHRLEVEL REFERENCE then the consistency of all of the objects is guaranteed if recovered to that copy.

- SHRLEVEL CHANGE allows the object being copied to be read or written to during the copy process. A SHRLEVEL CHANGE image copy is a ‘dirty’ copy. There is no guarantee of consistency if recovery is performed to this copy. All recoveries must be to a point of consistency, and so the log is needed when any recovery utilizes a dirty copy.

Concurrent copy

Concurrent copies are another form of image copies that use IBM system software products, in particular the DFSMSdss product, to create the copies. Concurrent copies may get you a higher level of availability because the amount of time it takes to logically copy an object is much shorter than the time to physically copy it. With a concurrent copy, the underlying object being copied quiesced, and then a command is issued to the system software to start a concurrent copy. Once the copy process is initiated, the object is fully accessible, and the copy continues against a logical image of the object as it looked when the copy was initiated.

One major advantage of this is that a SHRLEVEL REFERENCE copy can be taken very quickly allowing for a higher level of availability for a stable copy. The disk devices on which the objects reside must offer FlashCopy or the equivalent support (depending upon vendor) for you to be able to use concurrent copy.

Copy parallelism

Image copies can be created in parallel either manually or by using parallelism within the utility. Using parallelism can dramatically reduce the amount of time it takes to copy all of the objects in a database, and thus improve availability.
Manual parallelism involves placing objects or lists of objects into separate backup jobs, and then scheduling these jobs to run concurrently. This allows for a high level of flexibility and balancing of the jobs with each other and across multiple machines. It is also a manually intensive process that may require adjustments as objects come and go, or change in size.

Your utility software may be able to automatically parallelize the copy process. This is normally achieved by specifying the objects or lists as normal, and then setting a parameter indicating the level of parallelism. This option is definitely easier to maintain, but comes with some drawbacks. The jobs themselves may cause resource utilization problems if machine consumption changes during the copy process. Also, the parallelism is limited to the job executing, so manual parallelism is still required to utilize multiple machines. Finally, if your recovery strategy relies on a certain order of execution, then the recovery might be more difficult if the copy process is dynamic, as with automatic parallelism.

**LISTDEF**

To copy objects in a single utility execution, specify a list of objects as input to the COPY utility control statement. The advantages to specifying a list of objects is that the utility can take advantage of parallelism to process the objects quickly, and the list enables a set of related objects to be copied together. All of the objects in the list are processed together. They are drained together, and their backup information is stored at the same time in the SYSCOPY catalog table.

A LISTDEF is a utility control statement that defines a list of DB2 objects and assigns a name to that list. This list can be used by a variety of utility functions. The name associated with a list in a LISTDEF can be specified in the invocation of the subsequent COPY utility execution to process all of the objects in the list. The LISTDEF allows you to easily group objects together, and to use wildcards in identifying objects. This allows any utility execution to adapt to changes in objects without having to change the utility invocation. For example, you can specify to copy an entire database, and regardless of the changes in the database, all the objects will be backed up.

The TEMPLATE utility control statement defines data set allocations. This template can be used to generically name data sets that the COPY utility will define for each copy it creates. The combination of LISTDEF and TEMPLATE allows for a very flexible backup strategy that adapts to a changing database automatically.
Stacked data sets

The concept of stacking data sets applies only to image copies that are written to tape. This allows for a significant savings in physical tape media because multiple copy data sets can be placed as multiple files on a single or multiple tape volumes. Stacking can also save a significant amount of time in the creation of the image copies because tapes can remain mounted for the creation of multiple copies instead of a new tape being called for each copy. You can establish manually by using specific JCL in the backup job, or automatically by the utility software. You can use stacking in combination with utility parallelism (again, either manually by having separate jobs or by having the utility software do it) to dramatically decrease backup times.

While stacking and parallelism can dramatically reduce backup times, they can also have a negative impact on recovery times. The data set placement on a tape can affect the recovery time. If a data set is stacked on a tape, the tape device must locate that data set on the tape. If multiple objects on a single tape are recovered, they are best recovered in the order they are stacked on the tape. Otherwise, significant time will be wasted searching for the data sets. RECOVER utility software can use file sequence numbers from the SYSCOPY catalog table to properly call for tapes and locate the data sets on the tapes. However, this relies on coding of a list and associated jobs properly to properly allocate the stacked files on the tape. This is not always an easy task, and stacking is quite typically a backup advantage and recovery disadvantage.

Our recommendation is to copy to disk, and keep the most recent copy on disk in support of local recovery. DFSMShsm can be used to move and stack disk copies to tape as they age. Those copies can be copied or moved offsite for disaster recovery support.

Quiesce

The QUIESCE utility establishes a point of consistency for an object or set of objects. When a quiesce is taken, the relative byte address (RBA) or log record sequence number (LRSN) value associated with the quiesce is recorded in the system catalog and provides for a recoverable point in time. It can be really important to establish consistent points during the business day, especially if you are trying to establish a logical end of business day. The QUIESCE utility can help you do that. Quiesce points are absolutely critical if point-in-time recoveries are needed in situations in which SHRLEVEL CHANGE copies are in use. Specify WRITE(YES) with QUIESCE to force all data for all objects being quiesced to be written to disk before the utility completes and is the default. WRITE(NO) will not force the underlying writes, and may reduce the amount of time it takes to establish the quiesce.
In DB2 for z/OS version 8 and earlier, consistent recovery can only be to a consistent point in time. Establishing points of consistency using QUIESCE becomes very important when you are using SHRLEVEL CHANGE image copies, or employing some sort of advanced recovery techniques. With DB2 V8, you can recover to a point in time; however, if that point of time is not a quiesce point, you are basically guessing at the reliable point for data consistency. With DB2 9, the recovery finds a consistent point as close to the point in time specified; therefore, a quiesce point is not necessary (but still strongly recommended). From a business perspective, you still need to know the point in time you are recovering to.

Depending upon application activity and certain software settings, a quiesce point may be difficult to obtain. If a list of objects is specified, all of those objects need to be at a point of consistency to allow for the quiesce point to be taken. This means that all in-flight units of work must be committed before the quiesce is obtained. Achieving this point of consistency can be especially difficult for high volume applications using plans bound RELEASE(DEALLOCATE); the QUIESCE utility must drain all claimers on the objects being quiesced before the point of consistency can be obtained. For these reasons, DBAs must coordinate quiesce activity with application developers to determine how the business updates the data, what points in time are appropriate, and when a logical end of business day can be established.

You can also use QUIESCE with WRITE(YES) to force changed data pages to disk, which allows for external non-DB2 actions to be taken against those objects involved. In some cases, you may want external copies (disk level backups) as part of an advanced backup and recovery plan. The QUIESCE utility can be used as part of an advanced plan, and updates the recovery base RBA or LRSN of the objects being quiesced (HPGRBRBA field in the header page of each object), which is critical for log only type of recoveries that can be used along with external copies.

A standard procedure for running an external copy may look something like this:

1. Start the object(s) as read-only.
2. Run a QUIESCE WRITE(YES).
3. Run the external copy.
4. Start the object(s) as read-write.

The RECOVER utility recovers data, and can recover data to a current point in time (usually up to the point of failure) or to a prior point in time (PIT). The PIT can be either a copy or a point in the log.
Objects that can be recovered include:

- A table space
- A partition
- An index space
- A single page
- A data set

The recovery process will probably use a copy and/or log records to recover the data. Information about the copies and log records needed for a particular recovery comes from the SYSCOPY catalog table and the SYSLGRNGX directory table. In a typical recovery, one or more copies are applied, followed by log records to apply changes to data up to the desired PIT or point of failure.

**Recover to current**

Recovery to current is the most common type of recovery in production environments, and is typically issued after some sort of failure. Objects used during a recovery to current can be a system level backup or full image copy, and any applicable incremental copies. All log records created since the applied backups are also used for a recovery to current.

A recovery to current is used when hardware or software (non-DB2) has destroyed the data, or when there is a site disaster. Recovery to current is usually enabled by the RECOVER utility, which can automatically determine the resources required (copies and logs) to perform the recovery.

Once table spaces are recovered to current, indexes can be rebuilt from the table space data or recovered. The decision to recover an index depends on if a copy exists, and the need to reduce the CPU and temporary storage necessary to rebuild the index. The recovery on an index can happen concurrently to the recovery of its table space.

**Recover to a point in time**

PIT recovery is the most common type of recovery, and typically happens in production when an application has corrupted data. The recovery is to a point in time before the application run time - to remove the corrupted data (and any non-corrupted changes). Recover to a PIT in production is also performed when a failure has occurred, and the desired recovery is to a prior point of consistency or a logical end of business day. This is typical in disaster recovery situations. In test environments, a PIT recovery can be used to restore previous test conditions or to establish a stable baseline for test or development conditions.
Recover to a point in time

Options for a PIT recovery that does not involve the logs include:

- **TOCOPY** specifies an image copy data set to recover to. If the copy data set specified is an incremental copy, the last full copy is used and all incremental copies up to the specified incremental copy are applied.

- **TOLASTCOPY** recovers to the last copy taken, which can be a full or incremental copy. For an incremental copy, DB2 will apply the last full copy and all incremental copies up to the one specified.

- **TOLASTFULLCOPY** recovers to the last full image copy taken.

Most recovery utilities recover to a point of consistency, which forces you to use only copies using SHRLEVEL reference. If you use other options, you will be taking a risk of recovering to an inconsistent point in time or require extensive log processing to reach a point of consistency. If the copy specified is not consistent, you can be recovering to a point of inconsistency (and taking your chances), or the recovery software will find a point of consistency automatically (but will probably require the log to do it).

Options for a PIT recovery that do require the log include:

- **TOLOGPOINT** or **TORBA** applies image copies and log records to the PIT specified. The log point can be an RBA value (non-data sharing environment) or an LRSN (data sharing). Typically the PIT specified refers to a consistent PIT, which can be a SHRLEVEL reference image copy or a quiesce point. Other points (that can be obtained from SYSCOPY) include the start PIT of a LOAD or REORG utility. Depending upon your version of DB2 or your recovery utility software, you may be able to specify any PIT and the software will find a nearby earlier point of consistency. You can also specify a PIT that is at an inconsistent point and take your chances.

If you are in a data sharing environment and need to find a possible PIT, you can use approximate LRSN values to find a time that may apply. In other words, the LRSN can be converted into a timestamp. We have used this in situations where data has been corrupted, but no point of consistency was achieved before the corruption. Management wants recovery to be “as close to the corruption as possible.” Depending upon your recovery software, you probably need to specify an LRSN for the PIT. You can pull the nearby LRSN values from the DB2 console log. Then you can convert those LRSN values to timestamps to find a recovery point. Keep adjusting the LRSN until you get to the PIT desired, and then use that LRSN value for the recovery. Here is an example statement that converts an LRSN to a timestamp value:

```
SELECT TIMESTAMP(X'C1E10813AF32' || X'0000')
FROM SYSIBM.SYSDUMMY1;
```

The result of the above statement is ‘2008-01-31-08.12.36.962080’.
Recover objects concurrently

You can specify a list of objects as part of a recovery utility invocation. You can key this list yourself or use a LISTDEF. Parallelism can be employed using your RECOVER utility software, or you can manually control parallelism with separate jobs. Your level of concurrency, whether manual or automatic, depends greatly upon how the copies were taken. In addition, stacking of copy data sets on tape can greatly impact the parallelism during recovery. The most efficient parallel recovery would be a manual recovery if the copy process was done manually and in parallel, and the recovery matches the backup processing. This can be quite efficient, but is really difficult to manage. Your recovery software should be able to consider how the backups were taken when it automatically invokes recovery parallelism.

If multiple objects are specified in a list as part of the recovery utility execution, then the utility software can apply log records for all of the objects in the list concurrently. This can dramatically reduce the amount of time it takes to apply the changes to the objects as only one pass of the logs is required. It is important when performing a major recovery to balance the amount of time it takes to apply a copy with the amount of time it will take to apply log records. That way a balance can be achieved between the number of recovery jobs executing and the time it takes to mount and apply archived logs.

Copy impacts on recovery

Where the required image copies are located can have a dramatic impact on recovery times and recover parallelism. Copies on disk are obviously much faster than copies on tape. The stacking of image copy data sets on tape can have a dramatic impact on recovery performance. Your recovery software should be able to respect tape stacking order, even if it executes in parallel mode. If you are creating multiple recovery jobs yourself, you need to query the DB2 SYSCOPY catalog table to determine the recovery sequence.

If you take partition level image copies, you can recover both the parts and the whole table space (one part at a time). A single copy of a partitioned table space does allow for both partition level and full table space recovery. However, the recovery of the individual parts may be slower because the full non-partitioning image copy must be searched for the partitions’ pages.

The availability of copies can have a huge impact on performance of the recovery. Dual copies of objects, with one copy sent offsite, may not always be the best choice. The failure to locate a copy or a bad tape may result in a fallback to an earlier copy or a request to send a tape back from the offsite vault. The same situation applies to archived log data sets. Some shops actually send the second copy offsite. This could be a serious problem if the first copy is bad and is needed for a local recovery. It is best to make copies of copies for offsite storage.
Recover the DB2 catalog and directory

The DB2 catalog and directory are made up of table spaces and index spaces, and they can be subject to failures, such as a media failure. There may be instances where you need to recover these objects. While we hope to never have to recover the catalog and directory, we must be prepared for it.

You must follow a specific order when recovering the catalog data sets, due to relationships and dependencies in the catalog, as shown here:

1. DSNDB01.SYSUTILX
2. SYSUTILX indexes
3. DSNDB01.DBD01
4. DSNDB06.SYSCOPY
5. SYSCOPY indexes
6. DSNDB01.SYSLGRNX
7. SYSLGRNX indexes
8. DSNDB06.SYSALTER
9. SYSALTER indexes
10. DSNDB06.SYSDBAUT
11. SYSDBAUT indexes
12. DSNDB06.SYSUSER
13. DSNDB06.SYSDATABASE
14. SYSDATABASE and SYSUSER indexes
15. All other catalog and directory table spaces and indexes
16. Catalog indexes (user-defined indexes that have not yet been rebuilt/recovered)
17. System utility table spaces, such as Query Management Facility
18. Real-time statistics objects, application registration table/object registration table (ART/ORT), resource limit specification tables
The objects listed in step 15 can be recovered together. After the indexes for SYSDBASE and SYSUSER have been rebuilt, all remaining catalog and directory table spaces can be recovered in a single RECOVER utility statement.

If the first 15 are not recovered in the order specified, you will receive an error message due to the need for an existence of a dependent table space or index space.

When recovering the catalog and directory to a point in time, it is imperative that all user data is also recovered to the same point in time. If not an inconsistency may be introduced that will corrupt data and can cause DB2 to fail system wide.

**Disaster recovery**

If you assume the default DB2 disaster recovery plan of recovery to the last archived log shipped offsite, then you need to ship copies of image copies to offsite storage. In addition, you need to copy the archived logs and ICF catalog off site. The copies you take need to be coordinated with the system programming staff and their site disaster recovery plan. It is best if application data copies are made before the catalog and directory image copies, and the catalog and directory copies should happen before the ICF catalog export. The quantity of logs shipped offsite must be coordinated with the image copies that are being taken so that every object will be covered by a span of log after the image copy is applied.

Other things needed for an offsite recovery include offsite backups of the DB2 program libraries, and reports of copies and recoverability of objects that can be generated from SYSCOPY catalog queries or a REPORT RECOVERY utility execution. You also need a plan! An automated plan is best, and it is relatively easy to create one. See the automation section of this guide for an idea on how to do this.

**Backup and restore system capability**

Beginning with Version 8, DB2 provides a fast and flexible way to recover with a system-level backup and restore utility. This feature, which can be very helpful for disaster recovery and subsystem cloning, is implemented with two utilities: BACKUP SYSTEM and RESTORE SYSTEM.

The two utilities perform the backups and restores of volumes defined to a copy pool in Hierarchical Storage Management (HSM) and the Storage Management Subsystem (SMS). The copy pool is a set of SMS groups that can be backed up and restored in one command. This capability invokes a new HSM service to back up and restore volumes represented by a subsystem.
Every subsystem that will be invoking the utilities will define two COPYPOOLS:

- A COPYPOOL for data (database copy pool)
- A COPYPOOL for logs (log copy pool)

HSM and SMS will determine which volumes belong to the copy pool and are needed for the backup/restore utilities. The database copy pool should contain volumes of associated databases, and ICF catalogs. The log copy pool should contain volumes with BSDS, active logs, and ICF catalogs. It is a best practice to have the DB2 logs and their ICF catalogs in the same SMS storage group. The ICF catalogs for databases should be separate and reside with the data.

There are two methods for performing the copies and restores:

- A FULL copy copies the database and log copy pool. These copies can be used to recover the entire DB2 subsystem during a normal restart recovery or by using the RESTORE SYSTEM utility.

- A DATA ONLY copy copies only the database copy pool. These copies can be used with the RESTORE SYSTEM utility to recover the system to a point in time. This utility can be executed from any member in the data sharing group.

The RESTORE SYSTEM utility can be used to recover the system to an arbitrary point between system copies. It uses the backups made by the BACKUP SYSTEM utility. When you use RESTORE SYSTEM at a disaster site, the DFSMShsm environment must be restored to a point in time that is synchronized with the restore of the BSDS data sets. Using the DSNJU003 job, a conditional restart control record is created to truncate the logs at the desired point in time. This point in time can be used for each member of a data sharing group. When DB2 restarts, it will enter System Recover Pending mode, and every DB2 member of the data sharing group must be restarted. At this point, the RESTORE SYSTEM utility can be run and then DB2 needs to be stopped to reset the System Recover Pending mode. In a data sharing environment, all members must be stopped. Then DB2 can be restarted.

### Object-level recoveries

A system-level backup can be used as a recovery base for an object. When using a system level backup for an object recovery, the RECOVER utility invokes DFSMShsm to restore the data sets for the object from the system-level backup of the database copy pool. The ability to restore objects from a system-level backup with the RECOVER utility can be prepared by setting the subsystem parameter SYSTEM_LEVEL_BACKUPS to YES.
If the system-level backup resides on DASD, it is used for the restore of the object. If the system-level backup no longer resides on DASD and has been dumped to tape, the dumped copy is used for the restore of the object if you specified FROMDUMP option in the utility.

The output from LISTCOPY POOL and PRINT LOG MAP can be used to see the system-level backup information. Output from the REPORT RECOVERY utility can also be used to determine whether the objects to be recovered have image copies, concurrent copies, or a utility LOG YES event that can be used as a recovery base.

There are some restrictions for using the system level backups as a recovery base. If any of the following utilities were run since the system-level backup that was chosen as the recovery base:

- LOAD REPLACE
- REBUILD INDEX
- RECOVER from image copy or concurrent copy
- REORG INDEX
- REORG TABLESPACE

If this is the case, the use of the system-level backup is prohibited for object-level recoveries to a prior point in time.

**BMC backup and recovery utilities**

BMC Software provides a suite of backup and recovery management solutions that reduce or eliminate outages for backups and automate and speed local and disaster recoveries. Through intelligent automation, ensure continuous availability of your DB2 on z/OS data by:

- Reducing CPU costs with snapshot copy, conditional copy, and cabinet copy
- Improving availability by taking backups with minimal or no outages and recovering all data quickly
- Improving productivity by enabling recovery practice scenarios
- Mitigating risk by ensuring that you have the data you need to recover

Advanced Backup and Recovery

By Susan Lawson and Daniel Luksetich, YL&A

Overview ................................................................. 32
Snapshot/FlashCopy .................................................. 32
    Snapshot technology ........................................... 33
    DB2 and snapshot backup ...................................... 33
Recovery from a snapshot .......................................... 34
    Suspending the DB2 log ....................................... 35
    Snapshot with log transmission .............................. 35
Mirrors ................................................................. 35
    Global mirror/XRC .............................................. 36
    Metro mirror/PPRC ............................................. 37
    Split mirror ...................................................... 37
Log propagation ..................................................... 38
GDPS .................................................................. 38
Data sharing .......................................................... 39
    DB2 failure recovery ........................................... 40
    Coupling facility failure - structure recovery ............ 40
Overview

As a business grows, the risk associated with a system failure increases. At the same time, the requirements for data storage and processing expand with an increase in the size of the data store, an increase in transaction volume, and an increased expectation for availability. This means that while management expects reduced outages and faster recovery times, there is less time to take backups and establish consistency for more data in a busy system! The RPO increases and the RTO decreases. In some cases, management expects continuous or near continuous operations.

This increase in data and volume, along with the availability expectations, challenges traditional DB2 backup and recovery. For example, you can't take a SHRLEVEL REFERENCE image copy if you need high update availability. Have you ever tried to take a quiesce in a high volume read-write environment? We know of many shops where executing a QUIESCE utility can bring an online system to its knees.

These conditions and requirements are especially challenging for site disaster requirements. The traditional approach of disaster recovery, where copies are taken and shipped offsite, is no longer good enough from a daily operational perspective as well as from an RTO perspective.

Advanced backup and recovery is basically any backup and recovery solution that goes beyond the traditional DB2 based solutions. Some of these advanced solutions are DB2 based, but most are techniques that utilize resources beyond those provided by DB2. An advanced technique can use advanced hardware or software solutions, or a combination of both. The intention is two-fold: increase availability for normal daily processing and reduce recovery times. Don’t confuse advanced backup and recovery with high availability solutions. A high availability (HA) solution means continuous or near continuous availability, which requires very specialized solutions that involve modifications to application processes. HA is not addressed in this guide.

An advanced solution does not necessarily mean a complicated solution. Most advanced solutions may actually be simpler than traditional backup and recovery. Most of these solutions are site disaster solutions, and most of them are going to involve personnel beyond DB2 DBAs and system administrators. Typically, the additional people involved are the DASD management staff.

Snapshot/FlashCopy

“Snapshot” is used here to generically describe advanced DASD-based solutions from all the DASD vendors that can take a precise point in time image of logical DASD volumes at specific times. These snapshots can be logical or physical copies. With a logical copy, the DASD subsystem maintains a logical image of a volume and tracks changes. With a physical copy, the DASD subsystem can create a full copy of a volume in the background while the volume is being updated.
A snapshot involves two phases: initialization and copy. During the initialization phase, the image of the source volume is preserved. During the copy phase, changes to the source volume are not reflected in the target volume while the target volume is created.

Consistency of data is not maintained during a snapshot. Therefore, during the initialization phase the data on the source volume must be stable. In addition, if multiple volumes are involved in a backup, all data across those volumes must be made stable and copied together.

**Snapshot technology**

Snapshots (names vary from vendor to vendor) are an internal DASD hardware solution that is microcode-based. The snapshot process can be initiated with a DASD subsystem command or by using system software. The IBM DFSMSdss COPY command can initiate a snapshot in DASD devices. Once the initialization phase of the snapshot is complete, the COPY command will terminate and the copy will be made by the hardware in the background.

When setting up for snapshot use for DB2 data, you need to make some plans. Organize DASD storage groups in a meaningful manner. Decide whether all application databases will be grouped together or will each application get its own DASD and snapshot-based plan. Separate system-related DB2 objects, such as logs, bootstraps, and catalog into their own DASD storage group.

There are vendor solutions available for “time correct” snapshots that allow for a set of volumes to be grouped and snapped at a PIT without interruption to DB2. As with any advanced solution, there are trade-offs and caveats. Some solutions are limited to a single DASD subsystem, and some require SMS management. Remember that a snapshot solution may fail, and you may not get notified of a failure. It is important to have an image-copy-based contingency plan.

**DB2 and snapshot backup**

The safest, fastest, and easiest way to back up your DB2 subsystem components and database data is to shut down DB2, dump all the DASD volumes, and then restart DB2. This assures a stable copy of everything. Of course, in a modern highly available environment you probably cannot get the outage time for such a backup, especially for large quantities of data where the dumps take hours.
Snapshot backups can get you all the advantages of the stability of a dump/restore technique without the large outages associated with such a strategy. This is because that while the physical backup of the DASD is in progress, the database can be up and available for read-write operations. The subsystem needs to be down, or in the very least in read-only mode, during the logical portion of the copy. The snapshot volumes can then be dumped for offsite storage.

The benefits of this technique are that the logical snapshot copy is much faster than image copies, and get you a consistent copy. No logs are required for a stable recovery, and the recovery time is shorter than using image copies. You must keep in mind, however, that the snapshot backup is a volume-level solution and not an object-level solution. It is also not a replacement for image copies. The snapshot copy becomes the RPO.

Snapshot copies can be used for entire subsystems. Snapshot can also be used for stable applications that can be separated into their own storage group and have clearly defined recovery points to the last copy.

**Recovery from a snapshot**

A recovery from a snapshot can be done at the subsystem level or the application level if that application’s data has been separated into its own storage group. The volumes involved are restored just as they would from a normal pack dump.

Log only recovery is possible when restoring from pack dumps. Keep in mind that log only recovery relies on the HPGRBRBA value in object header pages. This value could be out of date for an object that is not updated often, and beyond the range of available logs. In those situations, manual intervention will be required to make the object available.

Each object in the subsystem has two LEVELIDs in its header page: the level ID and the previous level ID. These level IDs are used for down-level or up-level data set detection. They are a safeguard against external actions on data sets. That is, if you are using dump and restore techniques (or DSN1COPY, snapshot, or other techniques outside the control of DB2) to back up data, DB2 may check the level IDs and reject access to the object if DB2 thinks the object is down-leveled or up-leveled. The level IDs are updated based upon the value of the DLDFREQ ZPARM parameter, which also has an impact on the updating of the HPGRBRBA value. If you set DLDFREQ to 0, down-level detection is disabled. DB2 will not check for down-level data sets, so you may have fewer problems with restoring from a snapshot (but you could introduce log-only recovery issues). If you have down-level data set problems with a restore, you can use the REPAIR utility with the LEVELID option to reset.
Suspending the DB2 log

If you cannot take DB2 down to make a snapshot copy, then you could take a snapshot with DB2 operational if you can suspend the log. The process would be to leave DB2 running, suspend the log, take the snapshot, and when the logical backup completes resume the log. Use the DB2 SET LOG SUSPEND command to suspend the log. Issuing this command externalizes log buffers, updates the bootstrap data set with a high written RBA (or LRSN) value, and then suspends all update activity subsystem wide. Once the command has completed, a snapshot can be taken. Once the logical copy has completed, issue the SET LOG RESUME command to resume all activity.

Taking snapshot copies while DB2 is active results in “dirty” copies. Upon restore DB2 can be restarted, and during restart all inconsistencies should be resolved. Unlike a copy where DB2 is down, the logs will be required during restart.

In very active environments, it may be difficult to suspend the log. All updaters must commit for the suspend to occur, or it could time out. While the log is suspended there is a read-write outage. Applications that need constant write capabilities would need to deal with this outage, which can take several minutes in a large environment.

Snapshot with log transmission

Snapshots can be used together with log transmission to provide for a faster RTO than traditional DB2 recovery. With this technique, regular snapshots are taken of the entire DB2 subsystem and data and shipped to offsite storage. The DB2 logs are transmitted to offsite storage at regular intervals, typically at each archive log process. Sometimes the archive log process is forced at regular intervals (e.g., hourly), and the archives are transmitted.

On restore from the snapshot, the bootstrap data sets can be updated with the latest log information, the subsystem can be started without restart roll forward or backward, and then object level log only recoveries can begin. This is much like traditional DB2 disaster recovery without using image copies.

Mirrors

You can use mirrors in a number of ways, including global mirrors, metro mirrors, and split mirrors.
Global mirror/XRC

“Global mirror” is an IBM term for an asynchronous remote copy of DASD devices. With this solution, data is transmitted on a near real-time basis from a source set of volumes to a target set of volumes. One example of this type of copy is called Extended Remote Copy (XRC). There are equivalent solutions from other DASD vendors. This solution is hardware and software based, and requires a full set of remote DASD and an LPAR at a local or remote site because the System Data Mover software may be required (depending upon your vendor).

The advantage of this technology is that the changes at the DASD device level are transmitted from source to target at near real-time, and that time synchronization is maintained so that restores can be made to a time consistent backup. This means that the copy is protected against a “rolling disaster.”

This technology dramatically increases the RPO and decreases the RTO. It also dramatically increases costs as the technology requires additional machine power, network bandwidth, DASD, and potential remote site costs.

Rolling disaster

Consistency is everything! DB2 internally guarantees consistency, but when you begin to move outside of DB2 with your backup and recovery plan, consistency becomes a significant issue. With an asynchronous copy strategy, the issue of consistency becomes a system-wide issue. DB2 is continuously writing to DASD: user data, catalog and directory data, the logs, and BSDS data sets. These writes are propagated from the source DASD to the target DASD on a near real-time basis, and any failure that occurs probably won’t result in all movement of data to terminate at exactly the same instant. This is known as a rolling disaster. As data is written to source data, it may not be replicated to the target DASD in the same order. Any recovery from the target DASD must consider this possibility. All of the global mirror type solutions can automatically handle a rolling disaster by issuing a time-corrected restore of the DASD at the target. This may or may not be possible across multiple DASD subsystems. Check with your vendor.

Recovery using a global mirror

After a disaster has been declared, you may need to IPL the secondary site (where the copies were held). Then, depending upon your vendor software, you may need to perform a restore procedure to prepare the DASD and make any time corrections. If data sharing is involved, then coupling facility structures are cleaned up or initialized, and DB2 is restarted.
Upon DB2 restart, DB2 will attempt to return to a point of consistency. This is typical for a normal crash recovery. Objects may have been changed and cached in local or group buffer pools at the point of failure, and depending upon your DB2 version, DB2 may attempt to restore these objects. Objects may be placed in GRECP (group recover pending) and/or in LPL (logical page list) status. In these situations, a DB2 -START DATABASE command may be needed to resolve these issues. In rare cases, you may need to issue a DB2 RECOVER utility if the object inconsistencies cannot be resolved. As always, an image-copy-based contingency plan needs to be in place.

**Metro mirror/PPRC**

In this technology, like the global mirror, data changes are moved from a source set of DASD to a target set of DASD. However, with a metro mirror, the movement of data is synchronous. This hardware-based solution is controlled by microcode at the DASD subsystem level. No host LPARs are required locally or remotely. One example is the PPRC product from IBM.

The advantage to a metro mirror is that the replication of data changes from source to target happens in real time. This provides for the highest possible RPO with no loss of data in the event of a disaster. There technology has some disadvantages, however:

- The copy I/O must complete before the source I/O is marked as complete, which can significantly affect the performance of applications at the source.
- There is a limit to the physical distance between the source and target.
- No time synchronization is possible for data stored across multiple DASD subsystems without the purchase of additional software and possibly hardware.

Because no protection against rolling disasters built into this solution, metro mirror requires something else to make a usable backup.

**Recovery using a metro mirror**

This technology on its own does not provide for consistency at the target. It needs help from some additional technology and/or techniques. Software is available to group sets of DASD into consistency groups, which can provide for time correction at the copy. A geographically dispersed parallel sysplex (GDPS) can provide automation and time consistency. A split-mirror technique can be used with a metro mirror to create consistent copies.

**Split mirror**

The split mirror technique was originally developed by IBM and SAP as a way to use a synchronous mirror technology to create stable PIT copies of DB2 subsystems and
data. It is a hardware-based solution, but it requires some DB2 interaction. The advantage is that stable PIT copies can be obtained with a smaller read-write outage than with snapshot. You get a very high RPO and small RTO with a solution that is not as expensive as GDPS. The disadvantage is that it is still relatively expensive compared to a snapshot, because it requires at least three times the DASD along with remote site costs. Split mirror can be used as a local and/or site disaster solution, and it can be adapted to your needs.

A split mirror is used to take tertiary copies as stable copies of the backup. These copies are used in conjunction with a global or metro mirror. In a split mirror situation, the mirror between the source and target volumes is suspended (split), and a snapshot of the mirror target volumes are taken. The split of the mirror can be made at a point in time where the log is suspended. This can happen in a much smaller window than a snapshot with log suspension. The target volumes can then be copied and the mirror resumed. If snapshot technology is used to make the copy, the mirror can be resumed very quickly.

Split mirror copies can be used as a PIT recovery point for both global and metro mirrors. They can also be used together with transmitted logs for a log only recovery to the last transmitted log.

---

**Log propagation**

Log propagation is a technique that can be used with copy techniques to achieve a higher RPO without the costs associated with high RPO. A tool is required to transmit log records to a remote site as DB2 writes log data. The log tool can build the log records offsite, and it can generate updates to the bootstrap data sets. This enables a log-only recovery to be conducted at a hot site for a high RPO, typically to point of failure. A common relatively inexpensive technique is to use a snapshot copy to move data offsite and log propagation to recover to point of failure. While not as quick as a GDPS or other mirror solution, it is far less expensive.

**GDPS**

A GDPS is an IBM solution that combines hardware and software to achieve the highest RPO and smallest RTO. It is used in conjunction with a metro or global mirror, and provides for system automation to automatically restart at a remote location after a site disaster. It can be set up to provide for both local and remote recoveries seamlessly and automatically. The advantage to this technology is that system automation replaces many manual recovery efforts, time consistency is
assured by the GDPS technology so rolling disasters are avoided, and the highest level of availability is achieved. Some disadvantages are high cost, processing power required at the remote site, and high network bandwidth. Even with this advanced solution, you still need tertiary copies!

Data sharing

Data sharing is an architecture supported in a parallel sysplex environment where up to 32 DB2 subsystems (members) can share data (concurrent read/write access) within a group. This is supported by use of a coupling facility that provides the means for communication between the subsystems, coherency of the data, and concurrency of the users. A data sharing environment has an impact on recovery because multiple DB2 subsystems can be updating the same data. Additional components can experience failures that must be accounted for.

Object recovery

Each DB2 member in a data sharing group writes to its own log. If multiple members are updating the same data, the logs must be merged during a recovery. Each member has its own BSDS. Recovery in a data sharing environment is generally the same as a recovery in a single subsystem environment, with a few additional considerations.

Because each DB2 member writes to its own recovery logs and BSDSs, each member also needs to be able to read the logs and BSDSs of every other member in the group, so these objects must reside on shared DASD with appropriate access granted. This accessibility is necessary because a media recovery may require logs from multiple DB2 subsystems. A group restart requires access to the logs of all members in the data sharing group. The shared communications area (SCA) structure in the coupling facility contains information about all members’ logs and BSDSs. During the backward log-recovery phase of DB2 startup, each member updates the SCA with new log information, which each DB2 member reads during a recovery. The BSDSs contain a copy of the SCA. Each DB2 members’ BSDSs contain the same information that is held in the SCA because SCA can fail (because it is a cache in the coupling facility).

When a DB2 object needs to be recovered, you may need to merge the logs from several DB2 members. The LRSN provides common log record sequencing across members and is used to control REDO/UNDO records for data sharing. The LRSN is a six-byte value derived from the sysplex timer timestamp and based on a store clock instruction. By using this LRSN, DB2 can coordinate the log records from each member that was performing operations on the shared data and apply the log records in the appropriate sequence.
Avoid storing archive logs on tape in a data sharing environment because recovery time increases according to the number of members whose archive logs must be processed, unless you use virtual tape Storage (VTS). Depending on where you keep these logs, the recovery process can become a lengthy one. Minimize the number of archive logs needed by having large active logs, incremental copies, and frequent commits. Archiving to DASD is best, but if you must use tape, never archive logs for more than one DB2 in the data sharing group to the same tape.

During a recovery, DB2 accesses the logs of all DB2 subsystems in the group and merges the log records in sequence by the LRSNs.

**DB2 failure recovery**

In a data sharing environment when a DB2 subsystem fails, the modify locks held on shared objects are retained, and the other DB2 members remain active. When that DB2 is restarted, forward processing of the log begins from the unit of recovery of the oldest in-doubt UR and the oldest pending write from the virtual pool to group buffer pool (GBP). In data sharing, when a member updates a page a force-at-commit process is invoked to force the changed page to GBP. Retained locks are freed at the end of subsequent restart. A DB2 subsystem can be restarted on the same or a different z/OS subsystem. It is a best practice to have an automatic restart management (ARM) policy in place using the RESTART LIGHT option to bring up the failed DB2 subsystem quickly to resolve any retained locks.

**Coupling facility failure - structure recovery**

If a coupling facility fails with LOCK or SCA, or if there is a loss of connectivity to the structure, a dynamic rebuild of the lock or SCA structure is triggered. This rebuild is triggered only if the involved system exceeds the rebuild threshold in coupling facility resource management (CFRM) policy, or if an operator command is issued. When a rebuild is caused by a storage failure and the rebuild fails, a group restart is necessary. All DB2 members must be brought down, and a coordinated restart of all members must be performed to rebuild the SCA or lock structure from the logs. It is recommended that each coupling facility has enough room to rebuild the structures of the other coupling facility so that rebuilds can occur without a major outage.

Changed pages that were in the GBP when a coupling facility fails may be marked group buffer pool recover pending (GRECP) and require page set recovery. This recovery requires a LOGONLY recovery when no image copy is needed. A –START DATABASE command will start the recovery. The GBP checkpoint determines how far back in the log to process. If group buffer pool duplexing is used, the structure is automatically rebuilt in the surviving coupling facility. Enable DB2 managed group buffer pool duplexing to minimize any outages. For best performance, do not duplex the LOCK or SCA structures because these structures can be rebuilt fairly quickly.
Chapter 4 Disaster Recovery Planning

By Susan Lawson and Daniel Luksetich, YL&A
Overview ............................................................ 42
The disaster database ........................................... 42
Disaster recovery health queries ............................ 44
Generating jobs .................................................... 46

By Christopher Duellmann, BMC Software
BMC techniques for disaster recovery preparation ........ 46
  Disaster and local site recovery simulation ............... 47
  Disaster recovery estimation ................................ 48
  Disaster recovery evaluation ............................... 48
Overview

It seems that as storage increases and volumes increase, we employ fewer people to manage the databases. So while it becomes more critical to manage larger quantities of data, the number of DBAs to manage the backup and recovery plan for that data is reduced. So, with increased demand and reduced resources automation becomes important for effectively maintaining the plan. An effective backup and recovery plan requires several things to be in place:

- Applications and application priorities must be tracked.
- Recovery priorities must be established.
- You must be able to quickly adapt to changes in databases, applications requirements, and priorities.
- You must be able to create backup and recovery processes.
- You must have reporting capabilities, including recovery health checks and the ability to document the ever-changing backup and recovery plan.

With a high level of requirements, it is best to leave the major tasks to DB2. It is actually quite easy to do this by using some database objects, SQL, and possibly another programming language.

The disaster database

A disaster database is one set of tables that you can build within a DB2 subsystem or in a data sharing group for which you intend to automate some or all of your backup and recovery processes. This database is an extension of the DB2 system catalog, and will contain metadata about the objects in the subsystem. There are many ways you can establish a disaster database, and this is only one example. The disaster database can contain any number of tables, including:

- The application reference table holds key application information based upon the database name or names. It allows you to relate database objects to the various applications using DB2. You can set application-level priorities and assign personnel to roles within the disaster recovery strategy.

- The responsibility reference table can contain information about key personnel related to an application and their roles (for example, phone numbers so that key personnel can be contacted when situations arise, or when their application’s data has been recovered after a failure).
The recovery reference table, the heart of the disaster database, contains information about the local and site disaster recovery information for every object in a subsystem or data sharing group. It is keyed by the database name and object name so that it can be related to table spaces and indexes in the various DB2 system catalog tables. It also contains the recovery plan.

The backup reference table contains key backup strategy information for every object in a DB2 subsystem or data sharing group and relates directly to the recovery reference table. Every backup strategy is recorded here.

The recovery reference table can contain a good deal of information. However, the basic necessities are:

- Application ID. This is a foreign key to the application reference table and is used to directly relate objects to an application and all the associated information. This is optional as a minimal design does not need it.

- Database and object name. This information is critical, and comprises the table’s primary key. It is used to directly relate objects to their definitions in the DB2 system catalog, and can be considered an extension of the catalog. These fields are used directly to aid in checking the health of the recovery plan for an object, generating recovery jobs, and recording the recovery status.

- Access type. This field is an encoded value that indicates how an object is used. It identifies if an object is read-only or read-write. It is very useful when verifying the backup and recovery strategies for the object.

- Recovery method. This field is an encoded value that indicates how an object is to be recovered following a failure. This is key information in that it can be used to verify the recovery strategy for an object is in line with the backup strategy, and also be used to generate recovery jobs.

- Recovered flag. This field is an encoded value that is used after some sort of disaster is declared and a recovery is in progress. The flag is set as objects are recovered so that application personnel can track the status of their objects, and system automation can generate recovery jobs for objects yet to be recovered according to the priorities set in the application reference table.

While this seems like a lot of work and you may think you do not have the time, it really is fairly simple to get started. Minimally, you need a recovery reference table. Suppose that all of the indexes are rebuilt and that table spaces are recovered with DB2 image copy and log forward recovery. In that case you can populate the disaster recovery reference table with the following SQL statement:

```sql
INSERT INTO TREC_RECOVERY_REF
( REC_DBNAME, REC_TSNAME, REC_RECOVERY_TYPE)
SELECT DBNAME, TSNAME, ' ' FROM SYSIBM.SYSTABLESPACE;
```
Now the table is populated with the default recovery strategy for every table space. From this point on, simple updates can record exceptions to the default strategy and add application IDs. You can begin coding queries to check the system catalog for new table spaces or table spaces that need a copy that has not yet been taken.

The backup reference table can contain information about the various backup strategies for all objects in your subsystem or data sharing group. As with the recovery reference the information in this table is limited only by your imagination. Here are some example columns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application ID</td>
<td>This is a foreign key to the application reference table and is used to directly relate objects to an application and all the associated information. This is optional as a minimal design does not need it.</td>
</tr>
<tr>
<td>Database and object name</td>
<td>This information is critical, and comprises part of the table’s primary key. It is used to directly relate objects to their definitions in the DB2 system catalog, and can be considered an extension of the catalog. These fields are used directly to aid in checking the health of the backup plan for an object and how it relates to the recovery plan, and generating backup jobs.</td>
</tr>
<tr>
<td>Job name</td>
<td>This column is useful for grouping backups into jobs, and generating the jobs. It is part of the primary key.</td>
</tr>
<tr>
<td>Backup frequency</td>
<td>This column is useful information for reference, but can also be used as part of a job generation strategy.</td>
</tr>
<tr>
<td>Sort number</td>
<td>This column is useful if you are generating the backup jobs. It allows you to generate the backups in a specific order.</td>
</tr>
</tbody>
</table>

Disaster recovery health queries

Once the disaster database is created and populated in DB2, all the information about your backup and recovery plan is available online. This eliminates the need for paper documentation, or in the very least reduces that need significantly. The recovery plan is quite simple: the ultimate goal is to get the DB2 system back, recover the disaster database, and have the disaster database tell you what else to recover.
The disaster database, along with the DB2 system catalog, gives you a very powerful tool to generate backup and recovery jobs, determine the “recoverability” of objects, and manage priorities. Some example questions you can ask the disaster database are:

- What spaces exist?
- What spaces exist that don’t have a disaster plan?
- Is there a disaster plan for a non-existent space?
- For a recover to point of failure table space is there an image copy? Has there been an unlogged event?

The following query lists the table spaces that do not have a recovery plan. It is good for detecting when new objects have been added and no plan has been established.

```sql
SELECT A.DBNAME, A.NAME, A.CREATEDBY
FROM   SYSIBM.SYSTABLESPACE A
WHERE  A.NAME NOT IN
(SELECT B.TSNAME FROM
 DRP.RECOVERY_REF B
 WHERE A.DBNAME = B.DBNAME)
```

The following query lists table spaces with a plan that has been dropped:

```sql
SELECT REC_DBNAME, REC_TSNAME
FROM   QCPDBA.TREC_RECOVERY_REF A
WHERE  REC_TSNAME NOT IN
(SELECT NAME
 FROM   QCPDBA.SYSTABLESPACE B
 WHERE  B.DBNAME = A.REC_DBNAME)
ORDER BY REC_DBNAME, REC_TSNAME;
```

The following query lists table spaces that are expected to be recovered to current, but are currently not recoverable:

```sql
SELECT A.DBNAME, A.TSNAME, A.ICTYPE, A.ICBACKUP, A.ICDATE, A.ICTIME
FROM SYSIBM.SYSCOPY A, DRP.RECOVERY_REF B
WHERE A.DBNAME = B.DBNAME
AND A.TSNAME = B.TSNAME
AND (B.RECOVERY_TYPE = '1' OR B.RECOVERY_TYPE = '2')
AND NOT A.ICTYPE = 'Q'
AND A.TIMESTAMP =
(SELECT MAX(E.TIMESTAMP)
 FROM   SYSIBM.SYSCOPY E
 WHERE  E.DBNAME = A.DBNAME
 AND  E.TSNAME = A.TSNAME
 AND NOT E.ICTYPE = 'Q')
AND NOT EXISTS
(SELECT *
 FROM   SYSIBM.SYSCOPY C
 WHERE  C.DBNAME = A.DBNAME
 AND  C.TSNAME = A.TSNAME
 AND  (C.ICTYPE = 'F' AND C.ICBACKUP = 'RP')
 AND  C.TIMESTAMP =
 (SELECT MAX(D.TIMESTAMP)
```

---

Disaster recovery health queries
Generating jobs

Queries can be used as part of an automation process. We’ve created these processes using COBOL, REXX, and ISPF Edit PROCs and CLISTs. Jobs can be generated on the fly, or after any changes to the objects or plan. At the hot site, recovery jobs can be generated once the disaster database is recovered (the only recovery you have to personally remember). Once objects are recovered, they can be marked as such in the database, and the next recovery is generated according to priority.

Here is an example of a query that was embedded in a REXX routine used to generate a backup job using the image copy utility:

```sql
SELECT DBNM, TBLSNM, DSNUM, BKUP_TYP, OFFSITE_IND, JOBNM
FROM   PPOCSSR.DBBKUP
WHERE  JOBNM = ?
```

BMC techniques for disaster recovery preparation

Disaster recovery preparation is a challenging task. You do not run all jobs needed to restore a full DB2 subsystem at the recovery site every week to be well trained when the disaster happens. You probably do not even have all necessary jobs and data in place to be able to immediately respond to a system-wide outage. The recovery from a disaster is (hopefully) a rare event that requires a lot of skill, a plan how to perform it and the resources that have to be included to get the system back alive. And last but not least, nothing comes for free, meaning that the efforts you take choosing the right method and investing in an effective technology for an event that you wish it will never happen are costly and time-consuming.

Disaster recovery support is always a trade-off between costs and complexity versus data loss and recovery time. Some customers dump all DASD volumes periodically and accept the downtime of the system during this operation and a possibly data loss at the restore site. Other shops have implemented a mirror technology capturing DB2 log data in an ongoing process and transmitting it permanently to the remote site. This method ensures very little data loss just of in-flight transactions and humble recovery time but can be very cost-intensive.
The compromise at many companies is a technique to transfer all copies of application data and DB2 system images along with DB2 archive logs to the recovery site. In the event of a disaster, the DB2 subsystem recovery is made up to the most recent archive which can be from a few hours ago. This technique requires a complete set of non-trivial jobs and procedures to get DB2 up and running.

The BMC Recovery Management for DB2 solution provides full support of DB2 subsystem backup and recovery operation. It automates the process by generating all necessary JCL to perform a system backup at the local site and a system restore at the remote site.

Additionally, the BMC solution allows you to simulate the recovery procedure to make sure all essential resources are available, and to estimate the recovery time.

**Disaster and local site recovery simulation**

The recovery simulation feature is a helpful tool to actually test your recovery plan while reducing costs for complex testing scenarios. The BMC Recovery Management for DB2 solution offers this capability for disaster recovery and local site recovery simulations. It performs all recovery operations including reading the image copies and sorting and applying log records but does not execute the generated job streams, nor does it change any data. After the validity of all needed recovery resources has been verified, the output data sets are eliminated.

The JCL used for the simulation process is identical to the actual recovery JCL, except that it includes an option that indicates simulation mode. Generated recovery steps that would actually perform updates on the DB2 system are bypassed. You can use the simulation feature for any kind of local recovery, including full subsystem recovery.

Recovery simulation enables you to:

- Check for recoverability of the DB2 system resources and application data.
- Prove the availability and validity of all needed recovery resources.
- Test the log apply process.
- Improve the disaster recovery plan before running an expensive offsite recovery.
- Get an idea about the amount of time needed to do this type of recovery.
Disaster recovery estimation

The recovery estimation feature, available with the BMC Recovery Management for DB2 solution, can provide an estimate in hours, minutes, and seconds for the amount of time required to perform a complete disaster recovery, including system and application resources. Estimation helps you forecast an exact recovery time. By using this feature, you obtain a list of objects that take the most recovery time. Recovery estimation is faster than recovery simulation because it does not perform recovery operations.

For all simulated, estimated, and actual recoveries, BMC RECOVERY MANAGER for DB2 collects and saves statistical data. This enables you to analyze and compare different recovery scenarios and to evaluate and improve your disaster recovery process over time. The statistics allow you to determine if you are meeting service level agreements and to check alternative scenarios against total elapsed time for recovery.

Disaster recovery evaluation

The BMC Recovery Management for DB2 solution evaluates your disaster recovery procedure. Historical data of actual, simulated and estimated recoveries are automatically collected by the ARMBRDC and ARMBWDC programs (BMC RECOVERY MANAGER) and the AFRMAIN utility program (BMC RECOVER PLUS) and stored in repository tables.

The ARMBWDC program writes data collection information about the recovery start and end times for system resources and for DSNUTILB calls for application recovery. It can record information about the recovery phases, catalog and directory, catalog indexes, BMC RECOVERY MANAGER repository, and change accumulation repository.

The reporting program ARMBRDC consolidates the data collected during the recoveries in the UTILITY_RUN repository table and generates the recovery summary reports. It includes the ten objects with the longest recovery time from the actual recovery and shows the estimated time to recover each of those objects.

Data about disaster recovery simulations and estimations are stored at the local site where these procedures are performed. Data about actual disaster recoveries are collected at the remote recovery site where disaster recovery jobs are executed. This data must be transferred to the local site for analysis. The disaster recovery JCL that you use at the recovery site includes BMC Log Master for DB2 functionality to retrieve the desired data collection information which can be applied at the local site.
Data replication

People like to make copies of data. It is important to remember that a backup and recovery strategy is not a replication strategy. Trying to combine both generally creates more problems than the effort is worth. Various techniques and software are available from vendors to help move data from one location to another. This chapter summarizes some of these solutions.

Several vendor products designed specifically for replication are not covered in this chapter.

By Susan Lawson and Daniel Luksetich, YL&A
Data replication techniques .................................................. 50
Unload and load ................................................................. 50
Log analysis and replication .................................................. 50
DSN1COPY ........................................................................ 51
Snapshot/FlashCopy data set level copies .............................. 51

By Rick Weaver, BMC Software
Data replication using BMC technologies .............................. 51
Standard DBA Techniques .................................................... 53
Creative DBA Techniques ...................................................... 54
Choose the Right Technique .................................................. 54
Data replication techniques

You can use several techniques to replicate data, including:

- Unload and load
- Log replication
- DSN1COPY
- Snapshot/FlashCopy
- Vendor-specific techniques

Unload and load

This technique is quite simple and straightforward. You simply unload the data from the source and load it into the target. This technique provides the highest level of stability, and it works inside the database. However, it is probably the slowest technique, and it requires an outage at the source to achieve a point of stability.

Log analysis and replication

With this technique, log records are captured as changes are applied to data at the source location. To enable this type of replication, enable the DATA CAPTURE CHANGES option on the table in DB2 needs for the objects that are to be replicated. This is accomplished with an ALTER TABLE command. The amount of logging that will be done will increase with this feature turned on, but we have not seen measurable overhead even in the busiest of systems.

Log replication is almost always accomplished with the use of a vendor product. These products read the log with optional filtering for just the desired objects, and then generate SQL or load-formatted records to be applied at the target. The generated data can be applied at the target by running the SQL or loading the records. This technique typically requires some customization for your environment, which may involve creating load jobs or jobs that can run SQL statements, or perhaps even creating a custom program to merge the new data with the target-data based upon specific business rules. This technique also requires coordination with copy technology because an initial copy of the target must be created, and to refresh the target when the log replication fails.
The DSN1COPY program can provide a significant performance advantage versus the unload/load technique. With this technique, the underlying VSAM linear data sets are copied from the source to the target. This process happens outside the control of DB2, and the objects need to be offline during the copy, which makes it disruptive. In addition, DB2 internal identifiers need to be translated from source to target.

This technique also requires an understanding of how the DB2 linear data sets are named, and the impact of such things as partitioning, altering partitions, and online REORGs can have on the names. You also need to know the various DB2 internal object identifiers, what they apply to, and how to find them in the DB2 system catalog.

**Snapshot/FlashCopy data set level copies**

Some snapshot technology, including FlashCopy V2, allows for data set level copies to be taken. There are a lot of ways to move data using this technique, and like with DSN1COPY DB2 internal identifiers must be changed from source to target. It is important to remember that the objects being copied must be in a consistent state at the source during the logical phase of the snapshot.

At the target, the REPAIR utility can be used to set table space and index space object IDs in the header pages. The index OBID appears to be ignored by DB2, so it does not need to be changed at the target. The table OBID can be set at the target at table creation time to match the source.

This technique provides the same stability as the unload/load technique with the fastest performance and minimal outage at the source.

**Data replication using BMC technologies**

You sometimes need to replicate data from one DB2 subsystem to another DB2 system or RDBMS. You may want to update a QA environment with a copy of production data, or you may replicate or propagate data changes to a data warehouse.

Because replication frequently is done at an application or object level, granularity and flexibility in the extract process is important. Solutions aimed at capturing all transactions and replicating them to another site for disaster recovery failover do not address this challenge.
During the replication process, you can face additional challenges such as:

- Getting consistent data (no in-flight updates on the replicated output data set)
- Reducing or eliminating any outage on the source DB2
- Reducing resource requirements (CPU, I/O, and storage) to extract and apply the replicated data

Replication can be accomplished using native IBM DB2 utilities, service aids, or tools (DSNUTILB, unload, load, copy, DSN1COPY, Data Propagator), but consistency, availability, and resource utilization constraints are increasingly problematic.

Some of the large enterprise resource planning (ERP) applications (such as SAP on DB2 z/OS) have a subsystem-cloning requirement. In these environments, “cloning” implies that the DB2 data log and directory are copied, as well as the application data. That requirement is not the focus of this section. The focus here is on extracting a subset of DB2 data at an application level.

BMC Software offers several techniques to help replicate data from a DB2 z/OS source to a target system (usually also a DB2 z/OS system, but not always). The techniques build on the following products:

- BMC CATALOG MANAGER for DB2
- BMC CHANGE MANAGER for DB2
- BMC COPY PLUS for DB2
- BMC LOADPLUS for DB2
- BMC Log Master for DB2
- BMC RECOVER PLUS for DB2
- BMC SNAPSHOT UPGRADE FEATURE
- BMC UNLOAD PLUS for DB2
- Online Consistent Copy (OCC), a feature of the BMC Recovery Management for DB2 solution

This section presents BMC techniques (sometimes in conjunction with IBM tools) that at a higher level can aid DB2 z/OS data replication. This list assumes the presence of the native IBM DB2 utilities (DSNUTILB) and service aids (DSN1COPY). Because some operations can be conducted by either IBM or BMC utilities, the BMC brand name is specified on all BMC products. The use of specific BMC utility syntax or special function is noted in **bold**.
Standard DBA Techniques

The following recommended techniques use standard database administration practices for DB2 data replication.

1. DSNUTILB COPY or BMC COPY PLUS (both SHRLEVEL REF) on the source, followed by DSN1COPY or BMC RECOVER PLUS on the target (both may require OBIDXLAT)\(^1\)

2. BMC Snapshot Copy (BMC COPY PLUS and BMC SNAPSHOT UPGRADE FEATURE) on the source, followed by DSN1COPY or BMC RECOVER PLUS on the target (both may require OBIDXLAT)

3. BMC CHANGE MANAGER Data Only Migration, using DSNUTILB COPY or BMC COPY PLUS on the source, and BMC RECOVER PLUS OBIDXLAT on the target (this is an automated version of technique 1)

4. DSNUTILB UNLOAD or BMC UNLOAD PLUS (specify FORMAT BMCLOAD if using BMC LOADPLUS) on the source, followed by DSNUTILB LOAD or BMC LOADPLUS (specify FORMAT BMCUNLOAD if input is from BMC UNLOAD PLUS) on the target

5. BMC Snapshot Unload (BMC UNLOAD PLUS and BMC SNAPSHOT UPGRADE FEATURE, specify FORMAT BMCLOAD if using BMC LOADPLUS) on the source, followed by DSNUTILB LOAD or BMC LOADPLUS FORMAT BMCUNLOAD on the target

6. BMC CHANGE MANAGER Data Only Migration, using DSNUTILB UNLOAD or BMC UNLOAD PLUS on the source, and DSNUTILB LOAD or BMC LOADPLUS on the target (this is an automated version of technique 4)

7. BMC Log Master MIGRATE SQL on the source, followed by a SQL High Speed Apply engine on the target (target can be DB2 z/OS or any RDBMS such as Oracle on UNIX)

8. BMC CATALOG MANAGER Copy Table function (generates SQL SELECTs on the source, and INSERTs on the target)

---

1 A note on OBIDXLAT: For replication techniques based on image copy input, the source and target DB2 data sets will have internal identifiers (DBID, PSID, and OBID) that must match or be translated. It is possible to CREATE TABLE with OBID specification, and it is possible to create PSIDs within a database in the same sequence between two DB2 systems. It is technically possible to have DBIDs match between two subsystems if all of the objects in both DB2s are created in the exact sequence, or if the target DB2 is a clone of the source DB2 (the DB2 catalog and directory are copied from the source and used to define the target). However, it is nearly impossible to ensure DBID, PSID, and OBID all match between two non-cloned DB2 subsystems. So, for those techniques that state “may require OBIDXLAT,” unless the DB2 catalog and directory are cloned, it may be assumed that OBIDXLAT will be required.
Creative DBA Techniques

The following techniques make creative use of BMC recovery technology to aid in DB2 data replication.

9. **BMC RECOVER PLUS OUTCOPY ONLY** on the source DB2 (possibly coupled with Timestamp Recovery feature of BMC Recovery Management for DB2 solution to render a consistent output image copy from a SHRLEVEL CHANGE input), followed by DSN1COPY or BMC RECOVER PLUS INCOPY on the target (both may require OBIDXLAT)

10. **BMC RECOVER PLUS RECOVER INDEP OUTSPACE OBIDXLAT** on the source DB2 with target DB2 data set name output (possibly coupled with Timestamp Recovery feature of BMC Recovery Management for DB2 solution to render a consistent output with SHRLEVEL CHANGE input)

11. **Online Consistent Copy** (OCC, a feature of BMC Recovery Management for DB2 solution) on the source, followed by BMC RECOVER PLUS using OCC input with OBIDXLAT on the target

12. **Online Consistent Copy** (OCC, a feature of BMC Recovery Management for DB2 solution), followed by BMC UNLOAD PLUS with OCC input on the source, followed by DSNUTILB LOAD or BMC LOADPLUS on the target DB2. The unload file created with input from the OCC is a consistent sequential file. It could be ported to some other platform and loaded there, using appropriate utilities

Choose the Right Technique

This section presents some of the benefits and considerations of each of the techniques presented above, as well as field stories where relevant. When deciding which technique is best for your replication needs, consider which tools you already own. Choose a technique that works well in your environment and meets your targets for data consistency, source and target availability, and resource utilization.

**Technique 1: COPY on the source, followed by RECOVERY target (OBIDXLAT)**

**Benefit** Simple, can be supported by DSNUTILB COPY and DSN1COPY, or by BMC COPY PLUS and BMC RECOVER PLUS **OBIDXLAT**
Consideration This is a DB2-to-DB2 solution. Source and target object schema must match exactly (segmented table space to segmented table space, all table and column definitions must be the same). Consistent copy must be created using SHRLEVEL REFERENCE. This causes an application outage that can last several hours. DSN1COPY OBIDXLAT translation must be coded manually. BMC RECOVER PLUS can automate the OBIDXLAT. Both BMC COPY PLUS and BMC RECOVER PLUS are more efficient than the native utilities and use less CPU and elapsed time.

**Technique 2: BMC Snapshot Copy on the source, followed by RECOVER on the target (OBIDXLAT)**

**Benefit** Simple process using BMC snapshot technology on the source DB2, resulting in a very brief outage on the source DB2

**Consideration** Same as technique 1, except the availability impact of creating the consistent copy is drastically reduced. BMC Snapshot Copy essentially issues a QUIESCE for all the source objects, which flushes the buffer pools and establishes a recovery point. Once that is done, the objects are opened for read/write (R/W). BMC Snapshot Copy then uses one of several techniques to move the source pages to the output copy, making sure that any updates to the R/W objects are not put through until a “clean” version of the page is protected. The output copy always gets the clean pages, and therefore is registered as a SHRLEVEL REFERENCE copy in SYSCOPY.

**Technique 3: BMC CHANGE MANAGER, Data Only Migration (automated technique 1 or 2)**

**Benefit** Automates the build and maintenance of the data migration process, thus reducing the overhead on DBA time

**Consideration** BMC CHANGE MANAGER is a powerful tool to help with alteration and migration of DB2 schema as applications go through development cycles.

**Field story** A customer used BMC CHANGE MANAGER with BMC RECOVER PLUS for DB2 to synchronize data warehouse and query environments with production. BMC Snapshot Copy was used for input. The complete process was automated, synchronizing both structures and data migration with OBIDXLAT. This product combination enabled automatic resolution of the multitable table space OBID translation (a major advantage of using BMC RECOVERY PLUS). Further, the customer enjoyed fast and efficient recovery (including index builds) and dynamic allocation of all data sets, including multi-volume/multi-data set table spaces.
Choose the Right Technique

**Technique 4: UNLOAD on the source, followed by LOAD on the target**

**Benefit**  No need for OBIDXLAT processing; physical schema constraints allowed (e.g., segmented source to simple target table space allowed). Column definitions (sequence, data types) must match.

**Consideration**  Outage, CPU, and I/O on both source and target are required for consistent replication.

**Technique 5: BMC Snapshot Unload on the source, followed by LOAD on the target**

**Benefit**  A minimal outage on the source DB2 will create consistent unload data sets. Specifying `FORMAT BMCLOAD` reduces CPU time. If using BMC LOADPLUS, specifying `FORMAT BMCUNLOAD` reduces CPU time.

**Consideration**  BMC UNLOAD PLUS will need to quiesce all of the objects in the application before proceeding to create unload jobs. In a multiple-table partitioned table space environment, this can mean waiting for thousands of objects to clear. Further, the unload output data sets will be one per table space (even if there are 255 parts), which can lead to massive sort impact and excessive CPU.

**Technique 6: BMC CHANGE MANAGER, Data-Only Migration (automated technique 4)**

**Benefit**  Automates the build and maintenance of the data migration process, thus reducing the overhead on DBA time

**Consideration**  BMC CHANGE MANAGER is a powerful tool to help with the alteration and migration of DB2 schema as applications go through development cycles.
Technique 7: BMC Log Master MIGRATE SQL on the source, followed by SQL High-Speed Apply engine on the target

Benefit This technique works well for propagating source environment changes to a target. BMC Log Master browses the DB2 log, based on the filter criteria, and extracts all DML (INSERT, UPDATE, and DELETE) for a selected set of objects. The DML is copied to a file that can then be applied to the target DB2 via the SQL High-Speed Apply engine. This can be faster and cheaper than replicating the entire source application every time. It is especially useful for data warehouse applications. The BMC Log Master ongoing process ensures that the output file is consistent (only committed updates). In-flight updates are noted and carried forward to the next cycle. There is no outage to either source or target DB2 — all the work is done by reading log data and applying SQL. There is CPU and I/O consumption on both the source and target.

Consideration This technique assumes a one-time (or periodic) replication, using one of the other full-system replication techniques from the source to establish a base on the target system. The source and target table schema must match (column definitions must be the same across source and target). The physical structures can be different (for example, segmented table space to simple table space in DB2, or DB2 z/OS to Oracle).

Field story A large agency wanted to keep 60 days of data online to DB2. It extracted updates to production daily DB2 with BMC Log Master MIGRATE SQL Ongoing Process, and ported the data to update an enterprise data warehouse (EDW) via the High-Speed Apply engine. Data was pulled from 73 of 202 DB2 tables. After using this technique in production for 22 months, the agency replicated almost 23 billion rows of data to the EDW.

Field story A large European telephone company wanted to replicate data for disaster recovery purposes, but did not want to invest in expensive storage-based solutions. The company established a copy of the DB2 environment, and periodically transmits the SQL generated by BMC Log Master for application at a remote site. Based on the level of SQL activity, this technique was cheaper for the company to implement and sustain than a complete storage mirroring solution.

Technique 8: BMC CATALOG MANAGER Copy Table function

Benefit No outage on source or target DB2. This technique is great for moving a subset of data from one DB2 to another without replicating the entire application.

Consideration Generates SQL SELECTS on the source, and SQL INSERTs for the target. This technique generates CPU, I/O, and DB2 system overhead.
**Technique 9: BMC RECOVER PLUS OUTCOPY ONLY on the source, followed by OBIDXLAT on the target (possibly using Timestamp Recovery)**

**Benefit**
No outage on the source DB2. This technique makes use of normal image copy as input to replication process. This is a useful technique if the source and target DB2s are in different sites (or on different systems in the same site). The OUTCOPY ONLY syntax tells BMC RECOVER PLUS to use the existing source DB2 image copy as input; but instead of recovering the table space, the utility creates a new data set that can be ported to the target DB2. If the input image copy is SHRLEVEL CHANGE, the Timestamp Recovery process can be used on the source to render the output data set consistent (by applying log records up to a specified timestamp to the input copy).

**Consideration**
Both the source and target systems will experience CPU and I/O. An outage will be required on the target DB2 during the RECOVER OBIDXLAT process.

**Technique 10: BMC RECOVER PLUS RECOVER INDEPENDENT OUTSPACE OBIDXLAT on the source DB2, with target DB2 Data set Name output (possibly using Timestamp Recovery)**

**Benefit**
No outage on the source DB2. Uses existing image copy as input to replication. There is CPU and I/O consumption on the source DB2. There is an outage on the target DB2, but no CPU or I/O on the target system.

**Consideration**
This is a good technique when both source and target DB2 are in the same sysplex (or at the least on shared catalog and DASD environments). The INDEPENDENT OUTSPACE OBIDXLAT syntax tells BMC RECOVER PLUS to read the existing image copy on the source. Then, instead of recovering the source table space, the utility creates a new output data set with the naming conventions of the target DB2. OBIDXLAT is done on the fly. If the source image copy is SHRLEVEL CHANGE, then the Timestamp Recovery feature can be used to render the output data set consistent.

**Field story**
A large financial company bought a part of another business that was in DB2. The company wanted to bring this new application into its existing DB2 environment, which meant the entire subsystem would not be “recovered.” The company decided to create the tables in its DB2 subsystem, take the image copies from the new business, and do the BMC RECOVER PLUS INDEPENDENT OUTSPACE recover using OBIDXLAT. This enabled the company to quickly move the data to its new data center, while using fewer resources from both the old and new environments.
Technique 11: Online Consistent Copy on the source, followed by BMC RECOVER PLUS using OCC input with OBIDXLAT on the target

**Benefit**
No outage on the source system to create the OCC. There is little CPU or I/O to create the OCC (it is driven by storage data set snap technology). OCC renders the OCC consistent with log apply technology. The OCC is then input to the BMC RECOVER PLUS OBIDXLAT utility on the target, where replicated data sets will be created.

**Consideration**
OCC requires the appropriate storage technology. IBM, EMC, HDS, and STK all support data set snap capability and can be used by OCC. OCC is a feature of the BMC Recovery Management for DB2 solution. It cannot be unbundled; a full product license is required to use OCC. BMC Recovery Management for DB2 includes all the functionality of BMC COPY PLUS, BMC RECOVERY PLUS, BMC SNAPSHOT UPGRADE FEATURE, BMC Log Master for DB2, and BMC RECOVERY MANAGER for DB2, as well as exclusive solution features such as OCC and Timestamp Recovery. Per data set, OCC is very fast. Most of the work is done by the storage device. However, there is a level of serialization to the process, so conducting OCC on a large number of data sets may take a while (there’s no outage, but process time can climb). There is an outage on the target system during the OBIDXLAT recovery. The recovery will require CPU and I/O on the target system. The source and target schema definitions must match. Specifically, the table space, table, and column definitions must match (e.g., segmented TS to segmented TS).

**Field story**
A customer had a requirement to periodically replicate production data to a QA environment. No outage was allowed for source, and source CPU was constrained. BMC successfully installed and configured BMC Recovery Management for DB2 and EMC Snap to exercise the OCC function of the solution. We created a 4+M row table with an index, and replicated it from one subsystem to another.

The statistics:

- Before: Unload .45 CPU, Load .77 CPU
- After: OCC .03 CPU, R+ OBIDXLAT .13
- (86 percent reduction in CPU)

The OCC time is expected to remain somewhat flat, even for large objects (the bigger the source objects, the more impressive the CPU savings on the source environment). The customer is very happy with the results and will put together a program that lets application QA staff use the OCC-based replication at will, since it produces a consistent copy with no impact on source availability and little on resource consumption.
Technique 12: OCC followed by BMC UNLOAD PLUS, with OCC input on the source, followed by LOAD on the target

Benefit
No outage on the source system to create the OCC. There is little CPU or I/O to create the OCC (it is driven by storage data set snap technology). OCC renders the OCC consistent with log apply technology. The OCC is then input to BMC UNLOAD PLUS to create sequential output data sets. These can then be loaded into another DB2, using any DB2 LOAD utility, or ported to another platform for loading.

Consideration
OCC requires the appropriate storage technology. IBM, EMC, HDS, and STK all support data set snap capability and can be used by OCC. OCC is a feature of the BMC Recovery Management for DB2 solution. OCC is very fast on a per data set basis. Most of the work is done by the storage device. However, there is a level of serialization to the process, so conducting OCC on a large number of data sets may take a while (there’s no outage, but processing time can climb). The unload process, using OCC as input, requires CPU and I/O on the source system.

There is an outage on the target system for the load process. The load process will require CPU and I/O on the target system. The source and target table and column definitions must match.

Field story
A company has the following requirement and environment:

- Extract all of the data from a production DB2 source and replicate it to the target environment (not a DB2 system). The target environment is used for customer reporting purposes and must be refreshed daily.
- No outage for the source DB2.
- The extract file must be transactionally consistent (no uncommitted updates).
- Minimal impact on source CPU for data extract.
- Source data is a DB2 application with 24 DB2 tables, each in a partitioned table space, with 255 partitions (more than 6,000 data sets).
- Source data is approximately 3 TB (compressed).
- The target is on another platform. The ported file must be a sequential file.
- The extract and file format operation cannot exceed six hours. The extract file is then ported to the target environment and loaded there, allowing for next-day customer reporting.
Native utilities could not begin to meet the company’s “consistent with no outage” objective. While BMC Snapshot Unload could approach that objective, the snapshot unload process drove a massive sort that exceeded the CPU and process time parameters. *The combination of OCC and BMC UNLOAD PLUS created more than 6,000 consistent sequential data sets totaling more than 3T B within the company objective of six hours.*

All of the options included here assume that a consistent replicate is required. Some of the techniques require that the DB2 source be stopped to get consistency, while others have technology that can render a consistent replicate without an outage.

For clarification of the relative impact on both source and target in these techniques, see the following table.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Options</th>
<th>Source availability impact for consistent replicate extract</th>
<th>Target availability impact for replicate apply</th>
<th>Source resource consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IBM DSNUTILB COPY, IBM DSN1COPY OBIDXLAT</td>
<td>Yes (High)</td>
<td>Yes (Medium)</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>BMC COPY PLUS, IBM DSN1COPY OBIDXLAT</td>
<td>Yes (Lower)</td>
<td>Yes (Medium)</td>
<td>Lower</td>
</tr>
<tr>
<td></td>
<td>BMC COPY PLUS, BMC RECOVER PLUS OBIDXLAT</td>
<td>Yes (Lower)</td>
<td>Yes (Lower)</td>
<td>Lower</td>
</tr>
<tr>
<td>2</td>
<td>BMC Snapshot Copy, IBM DSN1COPY OBIDXLAT</td>
<td>Yes (Brief)</td>
<td>Yes (Medium)</td>
<td>Lower</td>
</tr>
<tr>
<td></td>
<td>BMC Snapshot Copy, BMC RECOVER PLUS OBIDXLAT</td>
<td>Yes (Brief)</td>
<td>Yes (Lower)</td>
<td>Lower</td>
</tr>
<tr>
<td>3</td>
<td>BMC CHANGE MANAGER Data Only Migration</td>
<td>As 1,2 above</td>
<td>As 1,2 above</td>
<td>As 1,2 above</td>
</tr>
<tr>
<td>4</td>
<td>IBM DSNUTILB UNLOAD, IBM DSNUTILB LOAD</td>
<td>Yes (High)</td>
<td>Yes (High)</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>BMC UNLOAD, IBM DSNUTILB LOAD</td>
<td>Yes (Lower)</td>
<td>Yes (Lower)</td>
<td>Lower</td>
</tr>
<tr>
<td></td>
<td>BMC UNLOAD, BMC LOAD</td>
<td>Yes (Lower)</td>
<td>Yes (Lower)</td>
<td>Lower</td>
</tr>
<tr>
<td>5</td>
<td>BMC Snapshot Unload, IBM LOAD</td>
<td>Yes (Brief)</td>
<td>Yes (Medium)</td>
<td>Lower</td>
</tr>
<tr>
<td></td>
<td>BMC Snapshot Unload, BMC LOADPLUS</td>
<td>Yes (Brief)</td>
<td>Yes (Lower)</td>
<td>Lower</td>
</tr>
<tr>
<td>6</td>
<td>BMC CHANGE MANAGER Data Only Migration</td>
<td>As 4,5 above</td>
<td>As 4,5 above</td>
<td>As 4,5 above</td>
</tr>
<tr>
<td>7</td>
<td>BMC Log Master REDO SQL gen from Log</td>
<td>No</td>
<td>No</td>
<td>Medium</td>
</tr>
<tr>
<td>8</td>
<td>BMC CATALOG MANAGER Copy Table SELECT/INSERT SQL</td>
<td>No</td>
<td>No</td>
<td>High</td>
</tr>
</tbody>
</table>
### Choose the Right Technique

<table>
<thead>
<tr>
<th>Technique</th>
<th>Options</th>
<th>Source availability impact for consistent replicate extract</th>
<th>Target availability impact for replicate apply</th>
<th>Source resource consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>BMC RECOVER PLUS OUTCOPY ONLY, DSN1COPY OBIDXLAT</td>
<td>No (if Timestamp Recovery is used)</td>
<td>Yes</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>BMC RECOVER PLUS OUTCOPY ONLY, BMC RECOVER PLUS INCOPY OBIDXLAT</td>
<td>No (if Timestamp Recovery is used)</td>
<td>Yes</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No (if Timestamp Recovery is used)</td>
<td>Yes</td>
<td>Medium</td>
</tr>
<tr>
<td>10</td>
<td>BMC RECOVER PLUS RECOVER INDEP OUTSPACE OBIDXLAT</td>
<td>No (if Timestamp Recovery is used)</td>
<td>Yes</td>
<td>Medium</td>
</tr>
<tr>
<td>11</td>
<td>BMC Online Consistent Copy, BMC RECOVER PLUS with OBIDXLAT</td>
<td>No</td>
<td>Yes</td>
<td>Medium</td>
</tr>
<tr>
<td>12</td>
<td>BMC Online Consistent Copy, BMC UNLOAD, IBM DSNUTILB LOAD</td>
<td>No</td>
<td>Yes</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>BMC Online Consistent Copy, BMC UNLOAD, BMC LOAD</td>
<td>No</td>
<td>Yes</td>
<td>Medium</td>
</tr>
</tbody>
</table>
This chapter discusses innovative backup and recovery techniques developed by BMC Software. Several methodologies are explained in detail, and BMC exclusive features are discussed.

By Christopher Duellmann, BMC Software

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>64</td>
</tr>
<tr>
<td>Objective of advanced technology</td>
<td>64</td>
</tr>
<tr>
<td>BMC Recovery Management for DB2</td>
<td>65</td>
</tr>
<tr>
<td>Smart copy techniques</td>
<td>65</td>
</tr>
<tr>
<td>Hybrid copy techniques</td>
<td>65</td>
</tr>
<tr>
<td>Snapshot copies</td>
<td>66</td>
</tr>
<tr>
<td>Cabinet Copy</td>
<td>68</td>
</tr>
<tr>
<td>Data set level instant snapshot</td>
<td>69</td>
</tr>
<tr>
<td>Incremental index copy</td>
<td>70</td>
</tr>
<tr>
<td>A major issue: consistent image copy</td>
<td>71</td>
</tr>
</tbody>
</table>
Overview

With continuous growth of data stored in DB2 z/OS and the increasing reliance on DB2 applications that require extensive database structures, you need an appropriate strategy for backup and recovery completion. As applications become more critical for business success, your backup and recovery strategy must become more elaborate and accurate to make use of high-sophisticated technologies and advanced solutions. The strategy must match your company’s overall business continuity plan to meet the predefined service level agreements and to ensure non-disruptive business operations.

Ideally your backup and recovery plan includes guidelines how to save and restore data without failures and without a negative impact on data availability. You need both hardware and software to achieve this goal. You must develop plans for local and remote recovery sites. You might need to establish and generate different processes for each site.

Many organizations are looking for solutions that maximize availability of business-critical data across the IT enterprise. Because an online recovery with zero outage is not available yet, rapid recovery techniques become quite attractive.

Objective of advanced technology

Reducing or eliminating the impact of backup and recovery operations on business processes is an important requirement and the number one issue for many organizations. The criteria to select and implement a certain technology according to the specific business needs can differ between shops and may have different priorities. Consider several objectives when planning for local and remote site recoveries.

- Recovery point objective (RPO)
- Recovery time objective (RTO)
- Recovery geography objective (RGO)
- Data consistency
- Availability impact
- Performance impact
- Cost versus benefit
- Risk of solution failure

All of these criteria must be evaluated and related to your business continuity plan. Some will drive a backup and recovery plan that meets your objectives. Using advanced features in backup and recovery procedures enables you to bypass the limitations that exist with standard capabilities.
BMC Recovery Management for DB2

BMC Recovery Management for DB2 provides individual DB2 z/OS backup and recovery point products and exclusive technology, including intelligent automation and unique advisory functions. BMC Recovery Management for DB2 works equally well for very large and complex enterprise application management (EAM) environments and for traditional DB2 application installations.

The following components are included in BMC Recovery Management for DB2:

- BMC COPY PLUS FOR DB2 (includes C+/MODIFY technology)
- BMC RECOVERPLUS (includes R+/RESOURCE MAXIMIZER and R+/CHANGE ACCUM technology)
- BMC SNAPSHOT UPGRADE FEATURE for DB2
- BMC Log Master for DB2 (includes High Speed Apply technology)
- BMC RECOVERY MANAGER for DB2

Smart copy techniques

Your backup strategy should always match your corresponding recovery strategy. For example, copying DB2 data at low cost with high speed and minimal or no outage against the running application are important criteria that may affect your decision about which technique to choose.

The requirement to create consistent, clean image copies versus fuzzy copies will influence your copy policy, particularly when you need continuous availability. However, the pressure you face when you need to recover DB2 databases will drive your priorities. It is good practice focus on which method to pick to restore the data in the fastest possible way and with accuracy. Then you can deploy the copy operations that support your recovery strategy.

The following sections introduce image copy techniques that go beyond traditional ways how to backup DB2 data. BMC Recovery Management for DB2 offers innovative copy capabilities through exclusive features and the powerful BMC COPY PLUS that is part of the solution.

Hybrid copy techniques

Hybrid technology is a term frequently used in the automobile industry. It implies the usage of two different technologies to get the best qualities out of each, in this example a gasoline engine and an electrical battery.
Analogous to this, some customers with large DB2 installations have implemented a technique referred to as Hybrid Copy. The user benefits from the strength of certain BMC COPY PLUS features and reduced local recovery elapsed times. Therefore, DB2 object availability during the copy process and recovery outage is greatly improved.

Typically, Hybrid Copy manages two types of copy operations depending on the size of the DB2 objects: large databases are copied with Instant Snapshots using whatever intelligent DASD devices are available, and small- and medium-sized databases are copied into a Cabinet Copy (explained below) file or with a regular copy process.

Specify the following parameters to establish a Hybrid Copy process in BMC COPY PLUS:

- MAXTASKS controls the maximum number of subtasks BMC COPY PLUS is using during the copy process
- OUTSIZE separates the objects by size to drive them to a specified BIGDDN output data set when the threshold is exceeded and to COPYDDN for spaces smaller than OUTSIZE
- IXSIZE forces the process to bypass indexes for a copy if smaller than the specified number of pages
- DSSNAP activates the Instant Snapshot feature
- STACK CABINET drives all copies to be written to a single output data set (Cabinet Copy)


Snapshot copies

The BMC DB2 copy utilities produce image copies faster than the native utilities. In their basic form, the BMC copy utilities read the database and produce an output copy. The database can be online or offline and the output copy can be on tape or disk. Several copies can be produced with one pass of the database.

BMC SNAPSHOT UPGRADE FEATURE creates a “clean” copy while the database is online (available for updates). It increases data availability when used with supported BMC utilities to create snapshots by using these methods:

- Software snapshots (also known as cache-based snapshot) — exploit z/OS virtual storage
Hardware snapshots — exploit Intelligent Storage Device (ISD) software such as EMC BCV or IBM FlashCopy

Instant Snapshots — exploit ISD software such as EMC Snap or IBM FlashCopy

Software and hardware snapshots are also called traditional snapshots. A traditional snapshot allows the supported utility to process data while a database remains available for updates. When the snapshot process starts, the database takes a very brief outage to establish a point of consistency. At this point, BMC SNAPSHOT UPGRADE FEATURE starts to provide the data to the supported utility consistent with that point of time:

For hardware snapshots, BMC SNAPSHOT UPGRADE FEATURE uses intelligent storage to provide pre-image records from a “frozen” copy of the database to the utility.

For software snapshots, BMC SNAPSHOT UPGRADE FEATURE monitors write requests to the database for the data objects being processed. When a record is changed, BMC SNAPSHOT UPGRADE FEATURE stores a pre-image of the record in its software cache.

As the utility reads database records during its job, BMC SNAPSHOT UPGRADE FEATURE satisfies the read request of the utility with the pre-image from either the hardware device or software cache. In this manner, the data read by the utility for that database is as it existed when the point of consistency was established, while the source database continues to be updated.

Instant snapshots, which are exclusive to BMC, are significantly different from traditional snapshots. When processing an instant snapshot, BMC SNAPSHOT UPGRADE FEATURE uses the appropriate intelligent storage interface to create (snap) a copy of physical data on a storage device to a different location on the same device (or on another device within the same control unit or frame). A copy of the data remains on the storage device after the utility has finished processing the job. BMC SNAPSHOT UPGRADE FEATURE can snap, or reapply, this copied data back to the original location for recovery.

For DB2 instant snapshot processing, you can create a “clean” instant snapshot (which requires a brief database outage), or a “fuzzy” copy (no outage, but log data must be applied to recover the database to a consistent point).

Because instant snapshot copies are unique to BMC, they are not registered like normal copies. A non-BMC recovery would not recognize or use the instant snapshot input. The instant snapshot copies are registered in a BMC repository table called BMCXCOPY.
Cabinet Copy

Cabinet Copy, a feature of the BMC Recovery Management for DB2 solution, allows you to copy all the table spaces and/or index spaces for a specified output descriptor into a single data set called a cabinet file. The cabinet file looks a lot like a stacked tape, except the overhead of opening and closing each output data set is avoided. The overhead of doing an open and close can add three or more seconds per space to the run time. This can add up to a considerable amount of time when hundreds of spaces are involved.

Instead, the cabinet file is allocated and de-allocated only once, regardless of the number of objects that are copied to or recovered from the cabinet file. Therefore the Cabinet Copy technique can significantly reduce overall run time and CPU usage when copying a large number of spaces.

Consider using the Cabinet Copy feature for large ERP applications with hundreds or even thousands of small table spaces and index spaces.

Cabinet copies not only improve performance, but they also save resources:

- save disk space due to the efficient use of space within a cabinet file
- reduce the number of z/OS catalog and VTOC entries
- reduce time for multiple recoveries or copies of image copies (COPY IMAGECOPY)

Other characteristics that apply to cabinet copies include the following:

- You can write a cabinet copy to either DASD or tape, and it can span multiple volumes.
- The COPY IMAGECOPY feature of BMC COPY PLUS can be used to create standard DB2-formatted copies or duplicated into another cabinet file.
- BMC RECOVER PLUS automatically detects cabinet copies and uses them for recovery of any spaces that are part of the cabinet.
- You can produce a cabinet copy of compressed indexes without expanding them.
- Cabinet copies are registered in the BMCXCOPY (a BMC internal equivalent of the SYSIBM.SYSCOPY table) table only.
Data set level instant snapshot

Instant snapshot technology uses the data set level snap capability of intelligent DASD devices. Data set level snapshot copies that are supported at the hardware control unit do not require the I/O needed for a standard copy and therefore are completed in a fraction of the time.

BMC COPY PLUS and BMC SNAPSHOT UPGRADE FEATURE for DB2 are both components of BMC Recovery Management for DB2 and are required to create instant snapshot copies.

BMC SNAPSHOT UPGRADE FEATURE for DB2 supports the following hardware devices for instant snapshot processing:

- EMC TimeFinder
- IBM RVA
- STK SVA
- Hitachi FlashCopy
- IBM FlashCopy

When a data set snap is performed, a physical copy (VSAM linear data set) of the original DB2 data set is created and cataloged in the ICF catalog. Instant snaps are always full copies and require the copy output data sets residing on DASD with dynamic allocation. They are registered in the BMCXCOPY table.

You turn on instant snapshot capability using the DSSNAP option of BMC COPY PLUS (which defaults to “no”) for making standard DB2 copies. DSSNAP YES creates instant snapshot copies.

With DSSNAP AUTO a fallback strategy is provided for circumstances where an instant snapshot operation is not able to complete, for example if the necessary hardware support is missing. That means you do not have to know if the DB2 data sets you want to back up reside on an intelligent DASD device; the tool finds that out for you. To enable BMC SNAPSHOT UPGRADE FEATURE for DB2 to fall back to a cache-based software snapshot from a failed hardware snapshot, you must set up a software snapshot cache and some options within the product. Otherwise, a de-escalation to perform a standard SHRLEVEL CHANGE copy is processed.

If you need a standard DB2 local backup copy out of an instant snapshot, you can take advantage of the COPY IMAGECOPY feature provided by BMC COPY PLUS. This technique may support your disaster recovery preparation strategy duplicating copies on tape to be used at the recovery site.

The instant snapshot copy can be a clean or fuzzy copy. Specifying SHRLEVEL CHANGE combines the enormous speed of the hardware assisted snapshot with a non-disruptive behavior against the application because no quiesce will be performed.
The restoration of such copies is also nearly instantaneous, which can significantly reduce recovery time. BMC RECOVER PLUS uses instant snapshot copies for the forward recovery of table spaces and index spaces. For data migration including instant snapshots, use the OBIDXLAT translation feature.

**Post-copy migration procedures**

A hybrid copy results in a lot of storage being used for image copy data sets. This is valuable at time of recovery — disk processing is faster and allows for more parallel processing. However, storage is not free. To mitigate the cost of storage, you can implement a process where only the most recent copy is maintained on disk; the others are archived to tape when a new copy is produced, typically with HSM HMIGRATE processing. During recovery, if there is a failure or some problem with the current generation of disk-based image copy data, the BMC recovery utilities can fall back to a previous generation and issue the HRECALL. The recovery will take longer, but it will proceed.

Some additional cleanup may be needed for the BMCXCOPY table as instant snapshot copies age completely out. BMC COPY PLUS includes a feature called C+/MODIFY that allows you to clean up both BMCXCOPY and SYSCOPY entries with one step.

**Incremental index copy**

Making incremental image copies of indexes applies an exclusive feature imbedded in BMC COPY PLUS as part of BMC Recovery Management for DB2. DB2 does not support incremental copies of indexes. It is usually much faster to recover large indexes than it is to rebuild them. Customers can profit by the advantage an incremental copy of an index provides over a full copy in both the copy and recovery process:

- Copying only the changed index pages takes less time
- Reduces the amount of space needed to place the copy output data set
- Less DB2 log has to be applied recovering the index space
- Reduces the load on virtual tape systems

Summing up, incremental copies of indexes give you the benefit of faster recoveries and saves space and time. It is good practice to use the CUMULATIVE option of BMC COPY PLUS that allows you to merge a new incremental copy with the most recent incremental copy. This will reduce the number of incremental index copies to possibly only one depending on your merge strategy. The recovery process will become streamlined and easier to handle.
Additionally, you can create incremental index copies using the following features:

- Incremental copies of indexes can be put in a cabinet copy.
- Incremental copies of indexes can be encrypted.
- Incremental copies of compressed indexes can be made.

Because DB2 does not maintain modification indicator bits for indexes, the index pages must always be scanned to identify the changed pages. Incremental index copy operations use the READTYPE FULLSCAN option.

You can automate the copy process by specifying FULL AUTO or CHANGELIMIT for incremental copies of indexes in BMC COPY PLUS. This allows you to automatically control the types of copies (full or incremental) being made and to avoid copies if there were no changes.

If your image copy strategy includes incremental copies of table spaces with FULL NO, FULL AUTO or CHANGELIMIT parameter, consider creating incremental index copies at the same run by adding the INDEXES YES option.

Incremental index copies are always registered in the BMCXCOPY table and must be recovered by BMC RECOVER PLUS.

**A major issue: consistent image copy**

You may need to create consistent image copies of DB2 data where no updates were in-flight at the time the copy process completed. Consistent copies may be required if you want to migrate data to another DB2 object or to a different DB2 subsystem (data replication) via OBIDXLAT processing for testing, quality assurance activities, or data warehousing.

The big challenge in the world of DB2 business-critical applications is that making consistent copies requires discontinuity at some point to obtain only committed data. Utilities running in SHRLEVEL REFERENCE mode usually have to quiesce the table space or use similar mechanisms to suspend an application program. Business processes, running 24 hours per day, 7 days per week simply do not allow any kind of interruption.

**Online consistent copy**

Online Consistent Copy (OCC) technology, an exclusive feature of the BMC Recovery Management for DB2 solution, addresses this requirement by creating transactional consistent image copies of table spaces and index spaces without any outage against the application.
An Online Consistent Copy starts with an instant snapshot SHRLEVEL CHANGE of the space being copied. Then, BMC Log Master for DB2 functionality is utilized to identify any in-flight transactions at the point in time at the end of the copy. BMC RECOVER PLUS techniques are called to externalize any changes to the copy that are still in the DB2 buffer pool, and to back out any changes in the copy that were made by in-flight transactions. No outage or quiesce is required for the source space. It only takes a few seconds per data set to make an instant snapshot, regardless of the size of the space. This method requires ‘snappable’ DASD, such as FlashCopy or EMC Snap.

A further enhancement of Online Consistent Copy technology makes use of a regular DB2 SHRLEVEL CHANGE copy to externalize the changes and to roll back the in-flight transactions. It then creates a merged, consistent copy as a VSAM data set. This eliminates the need for hardware that assists in making instant snapshots. However, this type of an Online Consistent Copy is recommended only for small objects.

Use the OUTSIZE option to determine when an object is sufficiently large to be copied with instant snapshot instead of a standard DB2 image copy.

The benefits of the Online Consistent Copy technology, coupled with BMC log analysis capabilities and backout processing, include:

- Continuous availability of the source object
- Consistent copy of the source object
- Minimal performance impact on the source system
- Instant snapshots created in seconds
- No host CPU usage and I/O required for instant snapshots (reduced costs)
- Alternate method available for a large number of small objects (OCC as a VSAM data set)
- Online Consistent Copy can be unloaded by BMC UNLOAD PLUS
- OCC is a non-standard image copy but can be processed by BMC utilities (copy entry in BMCXCOPY table)

Consider using Online Consistent Copies not just for a TOCOPY recovery or a migration process but also for a point-in-time recovery to a RBA/LRSN or a recovery to current. BMC RECOVER PLUS supports forward recoveries of any kind using Online Consistent Copies and log apply, thus providing the benefit to use one backup for different purposes.
Offline consistent copy using copy and log

BMC provides an alternate way to create a consistent image copy with a minimal impact to the application. This technique requires a quiesce for a set of table spaces you want to copy or a subsystem-wide quiet point by submitting an ARCHIVE LOG MODE(QUIESCE) command. After the quiesce or quiet point, the consistent point-in-time can be used for a copy technique that does not constrain the running application from reading and writing DB2 data.

BMC RECOVER PLUS can be deployed working as a copy utility. By merging the most recent full image copy, usually a fuzzy copy, with incremental copies, change accumulation files, and the DB2 log that was externalized after the copy was created, BMC RECOVER PLUS writes the output to a sequential image copy data set instead of a DB2 space. All pages of the copies are read and updated if log records exist to apply the modifications. The result is an up-to-date image copy, registered in SYSCOPY as a SHRLEVEL REFERENCE copy if the specified point is a quiet point transactional-wise. Otherwise, a SHRLEVEL CHANGE copy entry will be made in the catalog table.

The BMC RECOVER PLUS OUTCOPY ONLY option with TORBA or TOLOGPOINT does not process a point-in-time recovery; it simply creates a copy as of an earlier point-in-time. The DB2 spaces are not touched in any way, nor is the normal DB2 access interfered. With OBIDXLAT translation, you may want to take advantage of this copy in migration processes between different subsystems. However, you cannot create a cabinet copy or initiate a data set level snapshot with OUTCOPY ONLY.

For business-critical objects, generate image copies more frequently to avoid long log apply cycles in the case of a recovery.
A major issue: consistent image copy
# Chapter 7 BMC Software smart recovery techniques

*By Christopher Duellmann, BMC Software*

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>76</td>
</tr>
<tr>
<td>Point-in-time recovery</td>
<td>76</td>
</tr>
<tr>
<td>Backward-to-forward recovery automation</td>
<td>78</td>
</tr>
<tr>
<td>In-flight resolution technology and timestamp recovery</td>
<td>79</td>
</tr>
<tr>
<td>Recovery avoidance</td>
<td>81</td>
</tr>
<tr>
<td>Object recovery avoidance</td>
<td>81</td>
</tr>
<tr>
<td>Conditional restart avoidance</td>
<td>82</td>
</tr>
<tr>
<td>Data-set-level instant restore</td>
<td>83</td>
</tr>
<tr>
<td>Automatic index recovery to rebuild</td>
<td>84</td>
</tr>
<tr>
<td>Dropped object recovery automation</td>
<td>85</td>
</tr>
<tr>
<td>Transaction recovery using undo and redo SQL</td>
<td>85</td>
</tr>
</tbody>
</table>
Overview

The overwhelming majority of recoveries are needed because of errors made by application programs or transactions feeding improper data into the system. Up to 80 percent of all unplanned downtime is caused by software or human failures. In these cases, the DB2 subsystem is operating normally but is processing bad data. In other cases, mission-critical data can be destroyed by a media failure. These types of recoveries are application recoveries that must be executed at the local site. In most cases, data can be restored by choosing a valid point-in-time that exists before the error happened.

The BMC Recovery Management for DB2 solution supports a wide variety of local recovery methods, including:

- point-in-time recovery
- backward to forward recovery automation
- timestamp recovery
- recovery to a log mark
- volume recovery
- index recovery automation
- dropped object recovery
- data set-level instant restore
- transaction recovery
- object recovery avoidance
- conditional restart avoidance

These methods are described in the following sections. The recommendations rely on best practices gathered at many DB2 shops with large and complex DB2 environments.

Point-in-time recovery

DB2 itself provides options that enable you to recover table spaces and recover or rebuild indexes to a previous point in time. However, when table spaces and indexes are not in a corrupted state, consider alternative solutions. BMC Software DB2 recovery products provide exclusive functionality to back out changes to DB2 objects to a previous point-in-time without requiring image copies.

The native DB2 Recover utility always processes the DB2 log forward, starting with DB2 copies as the basis of the recovery. Even if the amount of data to correct is minimal, every page must be read and written for those table spaces and index spaces involved in the recovery process and residing on copies.
In this case, performance becomes an issue when unnecessary work can be avoided by using smart point-in-time recovery techniques. The goal is to save time and resources while restoring data to a logically correct state.

The backout recovery strategy of the BMC Recovery Management for DB2 solution is significantly faster than traditional forward recoveries. It does not require image copies to perform a point-in-time recovery; instead, it backs out the log records to undo or redo only those changes that occurred between the selected point in time and the current point in time. Pages that have not been changed are not processed in the recovery process. This method returns the spaces and indexes to the required state without the overhead of restoring image copies, or rebuilding or restoring indexes. This backward recovery technique provides a powerful way to recover your data to any point in time and with superior performance boost of only a percentage of traditional forward recoveries.

BMC RECOVER PLUS backs out your data by simply adding the keyword BACKOUT to the recover statement. BACKOUT allows BMC RECOVER PLUS to use the logs only, in most cases just active logs. Tape mounts are completely avoided.

All you need to perform a backward recovery is a point in the log where there is no uncommitted work. To ensure that objects are recovered to a point of consistency, use these BMC RECOVER PLUS keywords: TOLOGPOINT LASTQUIESCE or TOLOGPOINT LASTARCHQ (most recent ARCHIVE LOG MODE(QUIESCE) command). Consider quiescing a table space just before a batch cycle, and have a recover job in place with the BACKOUT option to be prepared if the batch job fails.

You can use the same backout strategy to recover index spaces to a consistent point. Certainly with large indexes, it is dramatically faster to back out the unwanted changes by reading the log backward and writing only those pages of the index that are affected by log records. Using the BACKOUT facility for a point-in-time recovery of indexes provides two major advantages that have a big impact to efficiency in terms of improved availability and resource savings:

- There is no need to create image copies of indexes because the backward recovery is using the log only.
- You can avoid rebuilding the indexes from recovered table spaces that can be a long running process if they are very large.

BMC Software recommends that you create an image copy with OUTCOPY YES as part of the BACKOUT execution to be used in a later recovery caused by a media failure because a recovery-to-current is not possible after the index has been backed out. You can also plan for a BMC COPY PLUS job to run after the backward recovery, which can be a fuzzy copy of the index space in order to eliminate any impact to your applications currently running. Restarting a recovery job with BACKOUT option is also possible but you may consider not changing the TOLOGPOINT option. There is no need to rerun the entire recovery job in case of a failure.
If you use have a huge environment to manage, or if you use business-critical applications like SAP or Siebel, you can optimize your recovery strategy with the BACKOUT option to meet SLAs by establishing an efficient, fast, and reliable recovery process. Also, BMC RECOVER PLUS helps keeping application development and QA staff productive. Shops with many home-grown applications, like insurance companies and banks, typically have long test cycles before new programs are transferred into production systems. Complex test scenarios often require initializing data frequently at a new starting point of a different QA cycle. DBAs must often turn back DB2 test environments as requested by application programmers and the QA department. Another demand is that all data must be accurate at the same logical point-in-time where testing starts over.

With backout recovery, databases get back online faster. Backout recovery shortens the amount of time that data is unavailable and improves the productivity of IT staff involved in time- and resource-consuming operations.

Backward-to-forward recovery automation

Circumstances might prevent a BMC RECOVER PLUS job with the BACKOUT option from completing normally, for instance, if a LOG NO utility was executed between the current time and the desired recovery point. In those cases, it would be helpful if the recovery process could automatically switch from backward to forward log processing without manual intervention.

The BMC Recovery Management for DB2 solution includes all the functionality of the single components like backout support for point-in-time recoveries. It also provides solution-exclusive features that automate the entire recovery process and makes it non-disruptive in most cases. The backward-to-forward recovery strategy in the solution detects situations where a backward recovery of a DB2 object is not possible. It automatically continues with a forward recovery of the object to the defined point-in-time.

Because backout recovery is the fastest recovery option in a point-in-time scenario, it will be considered first and is therefore the best choice.

When you specify BACKOUT AUTO, BMC Recovery Management for DB2 generates JCL for BMC RECOVER PLUS to perform backward recoveries for all eligible objects and forward recoveries for any objects that cannot be backed out. BMC Recovery Manager for DB2 automatically takes advantage of the ability of BMC RECOVERY PLUS to perform high speed backward or forward recovery, resulting in the fastest recovery possible.
In-flight resolution technology and timestamp recovery

To be prepared for a point-in-time recovery, you must establish a point of consistency to be specified as the target LRSN/RBA. The DB2 Quiesce utility can be executed to provide such a point in the log where all units of recovery are closed. This worked properly in the past, but can be impossible in high-transaction environments such as ERP applications where it is impossible to successfully execute a quiesce utility. In addition, quiesces cause unwanted overhead and data outages.

You can perform a conditional restart, which requires a recovery of the entire DB2 subsystem, even if only a subset of DB2 objects needs to be recovered. This method is unsatisfactory because you break a fly on the wheel.

It is your responsibility to avoid this problem by choosing a TORBA that is either the RBA of a SHRLEVEL REFERENCE copy or a QUIESCE, or one that corresponds to a point in the log where there are no in-flight logical units of work. BMC Log Master for DB2 can find such a point by examining a quiet point analysis on the DB2 log. If you could run a point-in-time recovery, even if a quiesce point was not established before, you would avoid the quiesce all together.

The BMC Recovery Management for DB2 solution provides such a feature to avoid a quiesce—which is actually an availability feature rather than a recovery feature.

The in-flight resolution technology, imbedded in the BMC Recovery Management for DB2 solution, enables you to perform a consistent recovery to any timestamp or LRSN/RBA. The ability to resolve in-flight units of work at any point in time or to any log point completely eliminates the need to execute quiesce utilities to establish consistent recovery points during application execution. Avoiding a quiesce can dramatically improve the availability of your DB2 data.

In-flight resolution technology is used by the timestamp recovery feature to perform a consistent point-in-time recovery to any user-specified timestamp. The BMC Recovery Management for DB2 solution translates the timestamp to a RBA or LRSN automatically. You don’t have to convert the RBA in non-data sharing systems manually. Uncommitted units of work are identified and resolved while recovering table spaces and indexes in both data sharing and non-data sharing systems. The BMC Recovery Management for DB2 solution supports this process to recover an entire group of DB2 objects or a set of groups.

The timestamp recovery feature can be performed as a backout recovery or a traditional forward recovery.
The backout strategy of a timestamp recovery includes all performance advantages over a conventional forward recovery that were already discussed in this chapter. It analyzes the updates within the time between the current point in time and the recovery point which again can be specified by a timestamp or by an RBA/LRSN. It then backs out these changes and all log records from in-flight LUWs. As the result, the spaces are returned consistently to their state as of the target time not including uncommitted updates.

The forward strategy of a timestamp recovery ends up with the same result starting with the most recent image copy before the specified target point. All log records except those from the in-flight unit of recovery IDs (URIDs) are applied up to the target point-in-time.

For almost all point-in-time recoveries, including the timestamp recovery feature, a physical backout is tremendously faster than a forward recovery and therefore the preferred technique. The advantages of the backout recovery technique using the timestamp recovery feature and in-flight resolution technology include:

- no need to read image copies
- indexes do not have to be rebuilt
- no tape mounts required
- no index sorts
- operation is simpler

The BMC Recovery Management for DB2 solution supports all features that come along with timestamp recovery and in-flight resolution technology including an efficient fallback strategy. Consider trying a backout recovery first for both table spaces and indexes. If the backout fails for any spaces, a forward recover is executed for the failed spaces automatically. When a backout of an index fails and no image copy of that index is available, the index is rebuilt.

Set these BMC RECOVER PLUS options as part of the BMC Recovery Management for DB2 solution to achieve the benefits of improved performance, greater availability, and higher automation:

- BACKOUT AUTO
- INDEXLOG AUTO
Recovery avoidance

The best recovery is the one that you do not run, so avoiding recovery is always the goal in complex recovery scenarios. If you must recover a large set of objects or—even more challenging—a complete DB2 subsystem, you can shorten the process if you would know what needs to be recovered and what can be skipped. Without intelligent analysis techniques, it can be tough to identify which objects have not been updated to avoid unnecessary recoveries. The BMC Recovery Management for DB2 solution provides an exclusive avoidance feature.

Object recovery avoidance

BMC RECOVERY MANAGER includes batch log range analysis and generates JCL to recover DB2 objects or the entire subsystem. By identifying objects that have not changed (XUNCHANGED keyword) between the current time and the recovery point, BMC RECOVERY MANAGER can avoid the unnecessary recovery of unchanged objects and can dramatically improve processing time. You can take advantage of the log range analysis feature in backup processing to reduce the overhead of unnecessary image copies (image copy avoidance).

The recovery avoidance technique can be used for local point-in-time recoveries of a set of spaces as well as for full subsystem recoveries (local point-in-time recoveries via a conditional restart). SYSCOPY and SYSLSRGNX information are analyzed to identify objects that appear to be unchanged. When a full subsystem recovery is performed, this information is stored in a DB2 global temporary table. After the DB2 catalog is recovered, BMC RECOVERY MANAGER compares information in the catalog with the information stored in the temporary table to verify which objects are actually unchanged.

For XUNCHANGED processing of indexes, the COPY YES attribute is a prerequisite.

Point-in-time recoveries for single DB2 objects and for a local recovery of an entire DB2 subsystem can greatly benefit from this functionality. Large ERP applications such as SAP often require the entire subsystem for recovery and backup processing. Excluding unchanged objects reduces the utility workload that must be performed and improves recovery performance. It also saves resources, including disk space for backups.
Conditional restart avoidance

Recovery of an entire DB2 subsystem has always been a big challenge and requires a conditional restart in most cases, followed by the recovery of DB2 system resources (DB2 catalog, DB2 directory, BSDS, archived logs, and so on) and an application object recovery. This type of recovery is not done often, and it requires a lot of experience to perform all necessary activities in the right sequence. With ERP environments consisting of thousands of table spaces and indexes per subsystem and several test and QA systems per application, it is impossible to perform a full subsystem recovery without help from tools.

Another issue is that applications such as SAP on DB2 z/OS create and drop objects dynamically by nature. In addition, referential integrity between DB2 tables is not specified in DB2, but logically present. Therefore, you cannot just recover a group of objects back to a point-in-time—you must recover the entire DB2 subsystem. It can be difficult—if not impossible—to get a quiet point in the log via the QUIESCE utility where all transactions are committed.

The BMC Recovery Management for DB2 solution fully supports the recovery of a full local subsystem with a conditional restart. It automates the entire process and generates all jobs necessary to bring the DB2 subsystem back to the point-in-time you specify with timestamp entry. If no ALTERs, DROPs, or CREATEs were performed in the subsystem between the recovery point and the current time, unchanged objects do not need to be recovered and a conditional restart is not required, Recovery time can be further reduced if a quiet point can be used as the recovery point.

The BMC Recovery Management for DB2 solution includes functionality to avoid a conditional restart whenever possible by providing a unique technique to dramatically shorten recovery time and simplify the whole operation. It analyzes the DB2 subsystem for unchanged objects, DDL activity against the DB2 catalog and alternate quiet points in the DB2 log. It then generates the most efficient recovery jobs possible including BACKOUT technology and BMC Log Master for DB2 functionality.

BMC Log Master for DB2 scans the log for alter, create, and drop activities reflected in the catalog between the specified recovery point and the current time. It also searches for quiet points within the user-defined recovery time frame.

The experience from the field shows that a conditional restart followed by the critical recovery of the DB2 system (DSNDB01 and DSNDB06) can be avoided in many cases. The entire recovery process is minimized to be as small as reasonably possible, and only an application object recovery is performed. It excludes unchanged DB2 objects and chooses the fastest recovery technique available.

The BMC Recovery Management for DB2 solution allows full control over the different recovery strategies with the CONDRESTART option to force the analysis to always pick the recovery method that is optimal for a particular case and has the best performance.
The CONDRESTART option for the batch program ARMBSRR controls whether the conditional restart method should be considered. You can choose the following options:

- AUTO (default) – instructs the program ARMBSRR to avoid a conditional restart if possible, but to include it if required. Depending on the DB2 release being used, a conditional restart can be avoided even if the catalog and directory must be recovered and no quiet point was found.

- YES – forces ARMBSRR to generate conditional restart JCL for the recovery, regardless of whether quiet points are available within the specified time range.

- NO – instructs ARMBSRR to avoid a conditional restart. If quiet points are found within the specified time range, the recovery is performed to the most recent point. If no quiet point is found, the recovery job ends with a return code 8 and message BMC80436E CATALOG AND DIRECTORY QUIET POINT NOT FOUND.

**Data-set-level instant restore**

BMC provides an instant restore recovery technique that supports forward recoveries by using instant snapshot copies. Instant snapshots are data set level copies of table spaces or indexes that are made by using intelligent storage devices. These non-traditional copies are registered in the BMCXCOPY table (not in SYSIBM.SYSCOPY) and are full copies only. Instant snapshots reside on disk and require some administration to fully catalog and archive them. Instant restore provides enough performance benefits to justify the administration efforts.

In an instant restore, the disk-based copy of a single data set is snapped back to the original data set. The restore time is very short, regardless of the data set size. Because the instant restore time can be calculated in seconds, it is an attractive recovery solution with BMC utilities. Native DB2 utilities cannot process an instant snapshot copy data set.

To make instant snapshot copies and recover with instant restore, you need the BMC Recovery Management for DB2 solution or all of the following products: BMC COPY PLUS, BMC RECOVER PLUS, and BMC SNAPSHOT UPGRADE FEATURE.

The input of an Instant Restore can be a clean or fuzzy instant snapshot copy. BMC RECOVER PLUS can apply log data to an instant restore process to conduct a full-forward recovery to current or to any point-in-time, thus making an instant snapshot copy of a data set useful for purposes other than recovery, such as data propagation.

Consider using instant snapshot and instant restore technology for very large table spaces and index spaces because the processes complete in seconds. Typically, business-critical applications require an instantaneous restoration of data in a recovery situation.
Taking instant snapshots for huge DB2 objects might be your first choice when establishing a recovery strategy that restores from a backup with zero I/O. This results in a huge reduction of recovery time.

**Automatic index recovery to rebuild**

When planning for an efficient recovery strategy for indexes, analyze the sizes of the indexes, measure the amount of time it takes to re-create them, and consider appropriate operations to separate the workload into several parallel tasks.

Recovering indexes from image copies and logs can be significantly faster than rebuilding them, and it makes the indexes autonomous from their underlying table spaces. It could be beneficial to recover table spaces and index spaces concurrently.

The BMC Recovery Management for DB2 solution enables automatic index recovery to rebuild. It identifies indexes that have copies and recovers them forward when possible. If no copy exists, or the index is not recoverable, the solution automatically falls back to an index rebuild.

The INDEXLOG AUTO option controls the execution of the RECOVER INDEX statement that first attempts an index recovery from image copies, the log, or both. However, if an image copy for the index does not exist, BMC RECOVER PLUS converts the request to a REBUILD INDEX request.

INDEXLOG AUTO also supports BACKOUT recoveries. If your recovery scenario includes point-in-time recoveries of table spaces and indexes using the automatic backout flavor (BACKOUT AUTO), BMC RECOVERY MANAGER automatically generates the INDEXLOG AUTO parameter to instruct BMC RECOVER PLUS to recover or rebuild the indexes. Therefore, the index copy will be considered by BMC RECOVER PLUS even in point-in-time recoveries that failed during the backward recovery phase and switched to a conventional forward recovery operation.

You may not want to create index image copies for all indexes in your installation regardless of their sizes. But to ensure that image copies exist, you can use the automatic index recovery to rebuild feature in combination with the automatic index backup feature IXSIZE, provided by BMC COPY PLUS. If you use the IXSIZE parameter to define the index copy threshold, BMC COPY PLUS creates index image copies only for those indexes whose size exceeds this threshold. This allows you to establish a standard in your shop how to create index image copies according to your SLAs without defining a separate rule for the individual object.
Dropped object recovery automation

This type of recovery is helpful in situations where a user accidentally dropped a DB2 object or an application program failed, eliminating data structures reflected in the DB2 catalog. Unless you have an alter or migrate policy that ensures that your data structures are archived on a regular basis, it can be difficult and time-consuming to re-create your DB2 databases and all dependent objects and rebind affected plans.

By scanning the DB2 log for information related to the dropped objects, BMC Log Master performs an automated drop recovery based on the information about the objects that do not exist in the DB2 catalog anymore. The type of recovery you request determines the information that is needed about the type of object (database, table space, table), the object name, and the time range when the object got dropped. You can specify the starting and ending point of a time frame by defining date/time or setting a log mark name or RBA/LRSN.

Several forms of output are generated to enable BMC Log Master and BMC RECOVER PLUS to automatically recover the dropped objects, including:

- Data definition language (DDL) to re-create the dropped objects
- BMC RECOVER PLUS syntax to recover the table space
- SQL statements to populate a recovered table with data
- Rebind statements for any invalidated application plans

You can request a drop recovery report that provides summarized information about the objects to recover.

Depending on the type of recovery you choose, BMC Log Master takes advantage of the following BMC RECOVER PLUS features:

- Can create output image copies during the recovery process
- INDEPENDENT OUTSPACE allows you to direct the output of the recovery to a space not used by DB2 for data migration purposes or test scenarios
- OBIDXLAT performs the translation of OBIDs to the recovered table space

Transaction recovery using undo and redo SQL

How can you recover when erroneous transactions ran against the database and, even worse caused subsequent updates to bring bad data into the system? Another type of recovery that uses the information stored in DB2 log records deals with the complexity of logical backout and recovery of application transactions.
Problem transactions can be difficult to identify and analyze. And you may want to keep the good transactions that were processed at the same time and affected the same rows as the transactions in error. BMC Log Master enables you to easily retrieve information from the DB2 log and provides several features that support you in backing out problem transactions, including:

- identifying and analyzing transactions in error
- creating UNDO SQL to reverse the effect of transactions in error
- creating REDO SQL to preserve valid transactions after a point-in-time recovery

Finding the most efficient backout strategy starts with the analysis of the problem transactions. Two reports support the analysis and help you select the right backout method:

- The Backout Integrity report compares the changes of interest against subsequent updates.
- The Summary report shows how many insert, update, and delete actions are included, and when an UNDO SQL or REDO SQL output file is generated.

Consider using the backout integrity checking feature and the SQL generation function for both UNDO and REDO SQL before deciding which strategy to use. With the Summary reports, you can calculate the volume of statements generated and compare the workloads for processing time estimation. You can also integrate updates resulting from referential integrity or trigger activity into the output file to reverse the effects of cascading delete actions or the nulling of foreign keys.

Executing UNDO SQL backs out the changes at a transaction level. Use the Backout Integrity report to determine which statements will be reversed by applying the generated UNDO SQL. The database can remain online during an undo transaction recovery. The effects of the selected bad transactions, specified by a filter and time frame settings, can be removed from the database without an outage and without losing valid work performed during the same time period. The undo of changes transaction-based is effectively an online recovery.

The REDO SQL process reapplies the valid transactions after having completed a recovery to a certain point-in-time before the undesirable changes. A REDO operation omits the bad transactions. Because you begin to restore transactions you want to preserve with the execution of a point-in-time recovery of related objects, the subsequent REDO SQL apply of all good changes can bring the table spaces forward to a current state, totally excluding problem transactions.

Be aware that:

- REDO SQL generation must be performed before you start the recovery.
- The filter and time frame you define reflect the changes you do not want to keep.
- The related table spaces and indexes will be offline during REDO SQL generation and the recovery to a consistent point.
The right solution for every backup and recovery need

By Rick Weaver, BMC Software
BMC Recovery Management for DB2 solution ................................................... 88
Which products/solutions are right for me? ......................................................... 90
Frequently asked questions ................................................................. 91
BMC Recovery Management for DB2 solution

BMC Software provides a comprehensive line of backup and recovery products for DB2 on z/OS. The product line includes individual DB2 backup and recovery point products that augment the native DB2 utilities with additional functionality and enhanced performance, and integrated, functionally rich solutions for all DB2 on z/OS environments.

Depending on your IT requirements, you can invest in individual point products, a combination of point products, or in the integrated solutions. Combinations of certain point products provide additional integrated functionality. For example, the combination of BMC RECOVER PLUS and BMC COPY PLUS will span the needs for backup and recovery for some shops. BMC COPY PLUS and BMC SNAPSHOT UPGRADE FEATURE provide another powerful set of functionality.

The BMC Recovery Management for DB2 solution integrates the point product technologies with additional intelligent automation and advisory features. The solution enables you to leverage the full breadth of available backup and recovery technology with a greater return on investment.

The BMC Recovery Management for DB2 solution provides all the tools you need to plan, simulate, and test for disaster recovery and application recovery. The scope of a recovery can be a group of transactions within an application, a single application, a group of applications, or for part or all of the DB2 system.

The BMC Recovery Management for DB2 solution integrates the following technology and provides a single point-of-control over backup and recovery processing:

- BMC COPY PLUS for DB2 (includes C+/MODIFY technology)
- BMC RECOVER PLUS for DB2 (includes R+/RESOURCE MAXIMIZER and R+/CHANGE ACCUM technology).
- BMC SNAPSHOT UPGRADE FEATURE
- BMC Log Master for DB2 (includes High Speed Apply technology)
- BMC RECOVERY MANAGER for DB2
Additional features and functions delivered with the solution include:

- Exclusive in-flight recovery technology that eliminates the need for application quiesces (timestamp recovery)
- Exclusive backout/forward recovery automation
- Exclusive backup and recovery avoidance
- Exclusive disaster recovery data collection and analysis
- Exclusive online consistent copy technology
- Disaster recovery simulation and estimation
- Disaster recovery mirror management
- Automatic dropped object recoveries
- Encrypted image copies

BMC provides the following DB2 backup and recovery point products:

<table>
<thead>
<tr>
<th>Product</th>
<th>Base Functionality, Core Value Added</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC COPY PLUS for DB2 (Includes C+/MODIFY functionality)</td>
<td>High-speed copy of table spaces and indexes with wild carding, stacking, parallelism, auto-copy, and SYSCOPY maintenance. Can take advantage of intelligent storage to speed up copies and off-load host resources.</td>
</tr>
<tr>
<td>BMC SNAPSHOT UPGRADE FEATURE for DB2 (also supported via XBM for DB2)</td>
<td>Provides facility for producing SHRLEVEL REFERENCE copies with a very brief outage. You can avoid more costly conventional means of creating copies, reduce CPU, and have more flexible backup options. You may find success by using a “hybrid copy” technique whereby large objects are driving to instant snapshots, and smaller objects are copied normally.</td>
</tr>
<tr>
<td>BMC RECOVER PLUS for DB2</td>
<td>High-speed recovery of table spaces and indexes, exclusive physical backout, dropped resource recovery with OBID translation. This product provides significant enhancements to recovery processing, unparalleled speed with backout recovery, and flexible recovery strategies. It can take advantage of intelligent storage to speed recoveries.</td>
</tr>
<tr>
<td>BMC RECOVERY MANAGER for DB2</td>
<td>ISPF menu-driven system for defining application recovery groups for local and/or DR purposes. Generates backup/recovery jobs and validates recoverability of DB2 objects. Provides ISPF interface for transaction recovery and unchanged analysis to avoid backing up and recovering objects that haven’t changed. Enables you to plan and test for recovery.</td>
</tr>
</tbody>
</table>
Which products/solutions are right for me?

Because each installation is different, you have different needs than other customers. Review this section to help choose the right solution for a given environment:

<table>
<thead>
<tr>
<th>Product</th>
<th>Environment Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Core point products:</strong></td>
<td></td>
</tr>
<tr>
<td>&gt; BMC COPY PLUS for DB2</td>
<td>Shops with large objects, many objects, many DB2 systems, heavy update traffic, and minimal maintenance window. Some of the techniques supported by this set of products allow for copy and recovery in seconds rather than minutes or hours.</td>
</tr>
<tr>
<td>&gt; BMC RECOVER PLUS for DB2</td>
<td></td>
</tr>
<tr>
<td>&gt; BMC SNAPSHOT UPGRADE FEATURE</td>
<td></td>
</tr>
<tr>
<td>BMC Log Master for DB2</td>
<td>Shops that need to report on DB2 transactions (for example, which user performed what process on what tables), need to undo changes at a transaction level with no outage (effectively an online recovery), or need to migrate data to another system.</td>
</tr>
<tr>
<td>BMC RECOVERY MANAGER for DB2</td>
<td>Shops that have many DB2 applications and systems to manage with a shared staff. By exploiting the menu-driven system, a small group of DBAs can support a large number of applications with efficiency and integrity. DB2 disaster recovery support is completely automated, as is validation of application group definitions and recoverability.</td>
</tr>
<tr>
<td>BMC Recovery Management for DB2</td>
<td>Shops that have a complex, shared DB2 environments and reduced experienced staff to support them. Robust and complete backup and recovery support supports high availability requirements across multiple applications sharing DB2 systems. The unique features of the Recovery Management for DB2 solution enable even inexperienced DBAs to recover an application to any timestamp, to perform application backups without the outage of a quiesce, and to automatically identify and perform the fastest possible recovery for application objects.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>Base Functionality, Core Value Added</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC PACLOG for DB2</td>
<td>Compresses DB2 archive logs up to 90%.</td>
</tr>
<tr>
<td>BMC Log Master for DB2</td>
<td>Allows the DB2 log to be used as a resource for reporting, analyzing, and logically extracting bad transactions via a high-speed UNDO. An easy-to-use interface allows you to perform routine audits, migrate changes, and recover applications on a transaction level. The High Speed Apply technology allows you to apply UNDO or REDO SQL that is generated by BMC Log Master on a highly managed, high speed basis. You can migrate data on a one-to-one basis between DB2 and Oracle, and you can migrate data to DB2 UDB.</td>
</tr>
</tbody>
</table>
Frequently asked questions

This section answers common questions about the BMC DB2 products and solutions.

Q. IBM has native utilities with DB2 that are “good enough.” Why should I invest in your utilities or solutions?

A. BMC has always augmented the IBM utilities to add functionality and improve performance. For instance, our customers challenged us to provide faster recovery to a point in time. We originally outpaced the IBM RECOVER utility by a factor of eight by implementing a “1960s technology” technique of sorting the log records into page/RID sequence and merging them with the image copy data. This allowed us to spin off the index keys and produce a new copy while the recovery was running. BMC RECOVER PLUS provides a unique physical backout recovery technique. In a recent recovery scenario, we recovered an application back to a desired point-in-time in less than a minute. Using normal recovery processing techniques (mount image copies and apply logs to the point-in-time), the recovery would have taken hours.

Q. Can I support the SAP, Siebel, or PeopleSoft applications with the IBM native utilities or do I need the BMC products?

A. With a great deal of knowledge, time, and expertise, you can use IBM native utilities for the large EAM applications, but it can quickly become a management nightmare. For example, a typical SAP implementation can contain more than 30,000 objects – some large, some small, some busy, some dormant, and some dynamic (meaning they are created and dropped as part of the application). It’s practically impossible for anyone but a very experienced DBA to properly recover everything, and it’s likely that even then some objects were recovered that might not have needed to be. Almost certainly, the recovery with native utilities would take far longer than one using the automated BMC DB2 tools.

Q. I plan to use a hardware snapshot for recovery. It seems simple. Wouldn’t that be good enough?

A. Perhaps. The hardware snapshot is especially attractive for disaster recovery support. However, you must STOP all processing or perform a SET LOG SUSPEND before you take the hardware snapshot. Some shops are reluctant to have even a brief outage – the typical hardware snapshot impact on availability is around 15 minutes. There’s also the storage management aspect of it – as your full-system application grows, it will eventually expand onto new volumes. Those must be added to the snapshot process, and included in the disaster recovery plans. Finally, if you want to recover to any point other than the hardware snapshot, you face a daunting task – one that may be impossible in certain situations. All things considered, it is always prudent to have a DB2 copy available for a normal recovery process.
Q. Isn’t the Backup System and Restore System in DB2 V8 good enough?

A. Only if you require the bare minimum of features and do not mind an enormous amount of planning and manual intervention.

Q. Why is it recommended by IBM and the EAM vendors that EAM applications execute in a stand-alone DB2 system? Can they execute in a DB2 system with other applications?

A. The EAM applications (such as SAP, Siebel, and PeopleSoft) tend to do things that are contrary to normal application processing. The biggest factor is that the relationship between tables is maintained by the EAM application, and the DBA typically doesn’t know anything about these relationships. This requires that DBAs always recover all the objects to the same point. The second biggest factor is the dynamic nature of objects which includes the CREATE and DROP of objects on the fly. This can play havoc with your backup and recovery strategy, especially in a recover to a prior point in time scenario. Usually the best way to handle the problems is to isolate the EAM applications and treat the whole DB2 system as an instance for the EAM.