Solutions for Large IMS™ Databases
# TABLE OF CONTENTS

- WHAT TO DO WHEN YOUR IMS DATABASES GET TOO BIG. ........................................ 1
- CONVERT FROM VSAM TO OSAM ................................................................. 1
- COMPRESS DATA. .......................................................................................... 2
- FAST PATH. ...................................................................................................... 3
- PARTITIONING ............................................................................................... 4
- CONCLUSION .................................................................................................. 9
WHAT TO DO WHEN YOUR IMS™ DATABASES GET TOO BIG

As the volume of IMS data grows, the need for more efficient space management and data access increases. If a database runs out of space, costly database downtime, recoveries, and unscheduled maintenance can have a significant impact.

While new IMS database development is scarce, existing IMS databases continue to grow. If you have web-enabled legacy IMS databases, you have probably seen significant growth in them. Traditional full-function databases are limited to 4 GB (VSAM) or 8 GB (OSAM). Once you reach those size limits, you have just a few options:

➤ Convert from VSAM to OSAM
➤ Compress IMS data
➤ Migrate databases to Fast Path
➤ Migrate databases to partitioning

The way you use the database can help you determine which path to take. This paper examines these options and provides items to consider.

Note: Adding more data set groups (DSGs) and moving some segments to another DSG can provide short-term relief for space constraints, but this is not a recommended option because it will require more I/Os.

CONVERT FROM VSAM TO OSAM

The simplest way to double the size of your IMS databases is to convert the access method from VSAM to OSAM. VSAM databases are limited to 4 GB; OSAM is limited to 8 GB.

To complete the conversion, allocate OSAM data sets, unload the VSAM data set, implement a DBD change to indicate that the database now uses OSAM, and reload to the OSAM data sets.

No application changes are necessary because the data structures and management by IMS remains the same. You may improve online application performance because of how IMS manages buffers for OSAM.

This method works until databases approach the OSAM 8 GB limit. When your database is that large, you must compress data or migrate to a new database type (Fast Path or partitioned).
COMPRESS DATA
Compression reduces cost by requiring less DASD for storing compressed data and reducing the load placed on I/O processing. Compression provides these benefits:

» Better buffer hit ratios for online transactions
» Smaller logs, which lead to smaller log volume, fewer log switches, fewer archive executions, and shorter change accumulation run times
» Dramatically lower costs for image copy space and storage because recoveries can be completed faster since compressed image copies can be applied quickly
» Improved utility maintenance times
  – For example, if it takes 30 minutes to reorganize a database with no compression, implementing a compression percentage of 50 percent can potentially shorten the reorganization to 15 minutes.
» Lower I/O costs
  – With compression, segments and data within buffers are smaller.
  – Because the data being read into the buffer is compressed, you can allocate smaller buffers or a smaller number of buffers.
» Ability to meet some security standards because compressed data is non-displayable
» Improved online IMS application response time through efficient use of IMS buffers and virtual storage
» Reduced segment splits and related I/O
» Efficient free space utilization
» Reduced elapsed time and I/O activity for sequential batch IMS applications

COMPRESSION TECHNIQUES
Choose a compression technique that yields the best compression percentage for your data. Several vendor compression tools are available, and the various compression techniques (algorithms) yield different compression percentages. Test your data to determine which compression technique works best for your environment. For example, the Huffman algorithm tends to provide the highest compression percentage for short, hierarchical data. On the other hand, the Ziv Limpel algorithm, which is used for hardware compression, works best with relational data and long segments.

HARDWARE VS. SOFTWARE COMPRESSION
For IMS, accessing and using native hardware compression is not a viable option. IBM® implemented hardware compression using a software compression product. Even if you choose hardware compression, you must implement software for IMS to communicate with the hardware.

COSTS AND BENEFITS
Compression provides many benefits, but it also has some costs, including the price of the compression software. (IBM provides some basic compression software at no additional charge.) Compressed data will also lead to higher CPU costs because data must be expanded and compressed whenever an application needs it. Before implementing compression, balance the cost of the software and CPU processing costs with the need for larger databases.

Compressed data must be able to be expanded. Vendors accept the responsibility for guaranteeing that compressed data can be expanded. Check with your vendors to see what data integrity checks they provide.

If compression does not resolve your database space issues, consider implementing partitioning or converting the database to Fast Path.
**FAST PATH**

If you have high application availability needs and compression did not solve your space problems, consider migrating to data entry databases (DEDBs). Fast Path processing, which includes the DEDB database type, was designed for high volume, high availability applications. DEDBs can have up to 2048 4 GB areas (total of 64 TB), offer improved I/O, and can have secondary indexes (with BMC Fast Path Enhanced Online Suite). In DEDBs, all data is stored randomly, so the only way to store data sequentially is to implement a sequential randomizer.

Note: If your data requires logical relationships or offline DL/I processing, you should not use DEDBs because IMS does not support these functions in DEDBs.

Fast Path supports high online transaction rates and delivers improved response times for database inquiries and updates. DEDBs can use expedited message handling that bypasses normal message queuing and scheduling.

Because Fast Path is an included feature of the IMS subsystem, every IMS installation has Fast Path available for their use. However, many IMS customers have never implemented Fast Path.

**DEDBS**

DEDBs are similar to an HDAM database and are stored in VSAM data sets called areas. A DEDB always stores all of the segments that comprise a record in a single area (data set). Each area has an overflow area to accommodate frequent update activity. This design offers flexibility in storing, accessing, and most importantly, maintaining self-contained portions of a database.

Fast Path supports databases larger than 4 GB by enabling database partitioning. DEDBs can have up to 2048 areas, and each area can have up to seven multiple area data sets (MADS).

**ADVANTAGES OF DEDBS**

DEDBs provide superior transaction throughput, manageability, large data volume capacity, speed, and lower overhead requirements than full-function databases. DEDBs process large volumes of data with a transaction rate higher than any other type of mainframe database. Fast Path DEDBs provide:

- MADS support to enable software duplexing of database areas
- Sequential dependents (SDEPs) for efficient journal-type applications
- Highly optimized I/O
- Virtual Storage Option (VSO), which enables a DEDB to be loaded into virtual main storage, thus improving performance because database segments do not have to be retrieved from DASD
- Two different locking mechanisms to reduce the amount of time an application waits for data. Normal mode is CI-level locking, which changes to UOW-level locking when an online utility is running.
- Shorter code path length for the application (up to 50 percent shorter)
- Reduced logging overhead during maintenance processes

Logging benefits include the following:

- “Syncpoint complete” when a database update is logged, which improves response time because a transaction verification message can be sent to a terminal before the update is written to DASD
- Database changes held in the buffer pool until committed, thus reducing logging overhead and eliminating the need for batch change backouts
- Eliminated blackouts because only the “after image” of the database is logged

Out of the box, DEDBs do not allow indexing. Indexes offer significant speed and resource usage advantages for IMS databases, and performance and convenience benefits for the applications they support. BMC Fast Path Enhanced Online Suite enables secondary index functionality.
PARTITIONING
Partitioning spreads database records across multiple partitions. Data set size limits apply to each partition, extending the capacity of the database as a whole. Partitioned databases offer performance improvements as follows:

» Parallel processing for routine database-management tasks, such as reorganizations and batch tasks
» Reduced data retrieval times for partitioned indexes
» Reduced I/O contention for database resources

You can implement partitioning with BMC Partitioned Database Facility for IMS, or you can migrate to High Availability Large Database (HALDB). Both options require a reorganization with a DBD change.

The cost of converting to a partitioned environment depends on more than just the cost of the tool. Look for hidden costs in the time involved if you must redesign databases to achieve partitioning. Consider the IMS and system resources required for the partitioned database. For example, partitioning tools can create additional data sets that must be managed and maintained, and that can require more space that in a non-partitioned database. Additional log records can also be an issue.

Your requirements for secondary indexes, logical relationships, particular database types, and performance requirements may favor one solution over another.

PARTITIONING OPTIONS
BMC Partitioned Database Facility for IMS provides a simple partitioning solution that can increase database size by a factor of 100. It fully supports secondary indexes (with no changes required), and it allows partitioned databases to maintain logical relationships to non-partitioned databases. BMC Partitioned Database Facility for IMS supports a variety of IMS database types. It supports, but does not require, DBRC.

A HALDB enables large databases to have all of the features of full-function databases plus an almost unlimited size. Partitions provide availability and maintenance benefits. For example, you can reorganize one partition while the remaining partitions remain online and available for processing.

The following chart compares the partitioning options.

<table>
<thead>
<tr>
<th>BMC Partitioned Database Facility for IMS</th>
<th>HALDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBRC supported but not required</td>
<td>DBRC required. BMC MAXM Reorg products provide a RECON Substitution Facility that enables you to use non-DBRC registered databases in a test environment.</td>
</tr>
<tr>
<td>Logical relationships allowed to non-partitioned databases</td>
<td>Logical relationships must be to other HALDBs</td>
</tr>
<tr>
<td>Secondary indexes can be partitioned</td>
<td>Secondary indexes must be converted to PSINDEX and must be loaded separately</td>
</tr>
<tr>
<td>No application changes required</td>
<td>Application changes are required to take advantage of partition independence and for applications that read secondary indexes (8-byte /SX field).</td>
</tr>
</tbody>
</table>

CONVERSION PROCESS
Converting to BMC Partitioned Database Facility for IMS is straightforward, and the process is similar to adding data set groups. No changes are necessary to primary or secondary indexes. The included utilities can recommend key ranges.

Converting to HALDB requires that you determine the size and number of partitions and how to distribute the data.
Determine the size of the partitions
While each of the 1,001 partitions cannot exceed 4 GB, one of the main advantages of HALDB is that you can design partitions so you don’t continually hit the 4 GB limit. If you choose to keep each partition no larger than 2 GB, then set the initial size to 1.5 GB and allow for growth. You can choose different sizes for each partition.

Your SLAs are important when choosing partition sizes. For example, if your SLAs state you need to recover 1 GB of data in one hour, factor this into your partition size planning.

Determine the number of partitions you need
It is important to design your HALDBs for maximum efficiency. While HALDBs can have 1,001 partitions, will you be able to manage that many partitions independently or in total?

Maintenance for HALDBs is managed at the partition level. HALDB partitions enable you to take one partition offline for reorganization, backup, and so on, while the other partitions are still available for processing.

However, there are times when you will need to process the entire database, for example, a reorganization to implement a structure change or a complete recovery. Before you migrate to HALDB, consider the implications of managing an extremely large database. Be prepared to manage the entire database at one time. Do you have a batch window large enough to reorganize or recover a 10 TB database?

Choose a manageable number of partitions so when you need to reorganize, back up, or recover the entire database, you can do so methodically and effectively.

Determine how to distribute the data
HALDB makes it easier to manage large databases because you can store data in the areas of the database that make the most sense for your shop. For maximum efficiency, carefully choose how to distribute your data. Consider what data is static and what data will change. The following examples show how you could choose to distribute data across partitions:

- A customer account database could be divided alphabetically (one partition for last names beginning with A, one partition for last names beginning with B, and so on). In this instance, it is likely that the data for partition Q will be much smaller than the data for partition S. Will having partitions of different sizes present issues for your shop?
- A billing database could be divided by time increments (all January activity in one partition, all February activity in one partition, and so on). For retail applications, the partitions for November and December may need to be larger than the partitions for other months.
- A database that is affected by different state laws could be divided by state (all Texas accounts in one partition, all Louisiana accounts in one partition, and so on).

Because you can distribute data as needed, you can also purge data easily. You can set up routines to purge all data in partition one when it meets your purge criteria (for example, when partition one is more than 12 months old). Because you no longer need to run separate purge jobs, you save time and resources.

Determine how HALDB affects your test and development environments
HALDB introduced a new level of complexity in development/test environments. In a full-function world, developers can each have their own copy of a database to work with and those databases are refreshed by simply copying existing data. This was enabled by not defining test/development databases to DBRC. HALDBs must be registered to DBRC because part of the database definition resides in the RECON data sets. You must choose among the following:

- Providing developers with separate RECONs
- Having all developers work off a single database definition, or
- Using full-function equivalents that are not registered to DBRC
Because the functionality and program return codes are nearly identical, most development/testing can be performed on a full-function equivalent, and in your system test environment you can use a HALDB to confirm that all is correct before moving the code into production.

The RECON Substitution Facility provided with BMC MAXM Reorg solutions enables you to have test databases that are not registered to DBRC.

DATABASE STRUCTURE AND PLANNING REQUIREMENTS

The types of supported databases, the number of data sets required, and supported database features can make a difference in the type of tool required to partition an IMS database. The following comparison summarizes the database types and features supported by BMC Partitioned Database Facility for IMS and HALDB.

<table>
<thead>
<tr>
<th>Support for</th>
<th>BMC Partitioned Database Facility for IMS</th>
<th>HALDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum data set size</td>
<td>4 GB for VSAM and KSDS</td>
<td>4 GB for VSAM and OSAM</td>
</tr>
<tr>
<td></td>
<td>8 GB for OSAM</td>
<td></td>
</tr>
<tr>
<td>Number of data sets per data set group</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Maximum number of partitions and data set groups</td>
<td>127</td>
<td>1,001</td>
</tr>
<tr>
<td>Secondary indexes</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Logical relationships</td>
<td>YES</td>
<td>YES (to other HALDBs)</td>
</tr>
<tr>
<td>Virtual paired relationships</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>HDAM databases</td>
<td>YES</td>
<td>YES (PHIDAM)</td>
</tr>
<tr>
<td>HIDAM databases</td>
<td>YES</td>
<td>YES (PHIDAM)</td>
</tr>
<tr>
<td>Root-only HISAM databases</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>SHISAM databases</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Index build</td>
<td>YES</td>
<td>YES (with BMC MAXM Reorg products)</td>
</tr>
<tr>
<td>Index per partition</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Index list data set (ILDS)</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Extended pointer</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Self-healing pointers</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Dynamic allocation</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>New IMS status codes</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>DBRC required</td>
<td>NO</td>
<td>YES, BMC MAXM Reorg products provide a RECON Substitution Facility that enables you to use non-DBRC registered databases in a test environment.</td>
</tr>
<tr>
<td>Partition independence</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Image copy indexes</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

Index Support

BMC Partitioned Database Facility for IMS supports both recoverable and non-recoverable primary and secondary indexes. You can restore the primary and secondary index using image copies and point-in-time recovery, or the index can be rebuilt from the database. The HALDB primary index (PSINDEX) and ILDS must be non-recoverable and cannot be image copied by IMS. Any procedures using IMS image copies of primary indexes will have to be changed with HALDB. The ILDS must also be considered for HALDB image copy or point-in-time recovery processes.

No changes are required for secondary indexes when you convert to BMC Partitioned Database Facility. The IDCAMS definition for a secondary index is the same as for non-partitioned databases. The size of the secondary index does not increase when converting the target database to BMC Partitioned Database Facility for IMS.
HALDB requires that secondary indexes be converted to HALDB PSINDEXes when the database is converted to HALDB. The size of the PSINDEX record is increased by the 28-byte extended pointer set (EPS) and the size of the root key. The /SX field, which is part of the key, will also increase from 4 bytes to 8 bytes. The size of the HALDB secondary index increases dramatically for HALDB.

Logical Relationships
Both BMC Partitioned Database Facility for IMS and HALDB support logical relationships and require that virtually paired logical children be converted to physically paired relationships. The logical parent concatenated key (LPCK) must be physically stored with the logical child.

BMC Partitioned Database Facility for IMS uses symbolic pointing to locate the logical parent. Pointer resolution is not required after the logically related database is reorganized. HALDB uses the extended pointer set (EPS) and self-healing pointers to update the EPS after a logically related database is reorganized.

BMC Partitioned Database Facility for IMS allows logical relationships to non-partitioned databases. HALDB requires that all logically related databases be in HALDBs.

Parallel Processing
Batch and BMP applications that read large databases can improve their performance by reading partitions in parallel. Both BMC Partitioned Database Facility for IMS and HALDB allow a database PCB to only read a selected partition by adding a JCL statement to the batch or BMP application. Running an application for each partition in parallel will greatly improve sequential read performance. The same can be done for batch or BMP programs that are driven by input files if the input can be separated for each partition to allow the programs to run in parallel.

Pointer Updates
HALDB self-healing pointers allow database or partition reorganization without updating the secondary indexes or logically related databases, which does improve reorganization performance. However, the pointers in indexes and logical relationships must be updated and healed eventually, and that normally happens when the first BMP with update intent reads the database from the secondary index or logical relationship.

The pointer will not be healed until an application with update intent can update the pointer. If your applications read the secondary index only with read-only BMPs, you must run an update BMP to heal the secondary index. Databases with a high-percentage of logical relationships between databases could experience performance problems with BMPs healing the pointers.

The number of locks will dramatically increase when the pointers are healed. The locks are not released until the application reaches a commit point. BMPs with update intent that do not take frequent checkpoints may hold so many locks that online performance can be affected.

Returning to a Non-Partitioned Format
Because of the increased size of the PSINDEX (up to three times the size of a non-partitioned secondary index) and the additional data sets and logging required by HALDB, returning to a non-partitioned database can be complex. Any application changes must also be backed out.

BMC Partitioned Database Facility for IMS adds no data structures to the system, and backing out the partitions is as simple as reversing the conversion process. A DBDGEN and ACBGEN will be required. Because no application changes are made to convert to BMC Partitioned Database Facility for IMS, there is no need to consider applications in the return.
Partition Independence

The most significant HALDB feature that BMC Partitioned Database Facility for IMS does not have is partition independence. With BMC Partitioned Database Facility for IMS, DBRC authorization is done for the entire database. HALDB DBRC authorization is done for each partition, and you can /DBR a partition to be used by database utilities while the other partitions are available online.

However, you might want to change your applications to take advantage of partition independence. DEDBs have always had area independence, and applications were designed to take advantage of the FH status code (area not available). Full-function applications were not designed for an equivalent “database record not available” status code.

An application does not know that a HALDB partition is unavailable until it makes a call to the partition and receives a BA status code. Most applications do not issue the INIT command for extended status codes, so the call to an unavailable partition will result in a U3303 pseudo-abend.

Archiving the Database

If you use database reorganization to archive and purge inactive segments, be aware that HALDB does not update secondary indexes during reload. Source segments cannot be deleted during reorganization because HALDB secondary indexes contain index records for source segments that have been deleted during reorganization, and deleting the source segments would result in an index pointer error. This design precludes the kind of common practice in which segments are deleted from the database during reorganization by a user exit or by modifying the unload data set.

Another archive approach is to keep the most current data in the first (or last) partition and then age the data by incrementing partition numbers during reorganization, with the oldest partition data being discarded or archived.

WHICH PARTITIONING OPTION IS RIGHT FOR ME?

Before moving to partitioning, consider the benefits (space and availability) versus the costs (maintenance of a very large database).

BMC Partitioned Database Facility for IMS may be the best option if you have:

» Logical relationships and don’t need to migrate all related databases to partitioning at the same time

» SHISAM databases that cannot be changed

HALDB may be the best option if you need:

» Partition independence (the ability to perform maintenance on selected partitions while the other partitions remain online and available for processing)

» Extremely large databases
CONCLUSION

When your IMS databases reach their size limit, you can change the database structure, implement compression, or both. Choose your option based on how you use the data and how much time you have to change it.

Start with the simplest methods. Change the access method for full-function databases from VSAM to OSAM, and implement compression. (You can do both of these at the same time.) If these don’t work, consider moving to Fast Path or partitioned databases. If you have some DEDBs already, converting other databases to Fast Path may be a good idea.

Converting from VSAM to OSAM and implementing compression do not require application changes, but they require a reorganization with a DBD change.

Converting to a DEDB may require a few application changes. For example, Fast Path requires a DLI checkpoint call at the end of processing. It is also a good idea to add certain status code checking. If your application requires logical relationships or offline batch processing, Fast Path is not recommended.

Converting to BMC Partitioned Database Facility for IMS requires no application changes. Converting to HALDB technically requires no application changes, but adding certain status code checking is recommended.

If you do choose to migrate to partitioning or DEDB, consider adding compression at the same time. This will give you the processing benefits of compression and also provide you even better space utilization.

Work with your vendors to implement the strategy that works best for you.

BUSINESS RUNS ON IT. IT RUNS ON BMC SOFTWARE.

Business thrives when IT runs smarter, faster and stronger. That’s why the most demanding IT organizations in the world rely on BMC Software across distributed, mainframe, virtual and cloud environments. Recognized as the leader in Business Service Management, BMC offers a comprehensive approach and unified platform that helps IT organizations cut cost, reduce risk and drive business profit. For the four fiscal quarters ended June 30, 2011, BMC revenue was approximately $2.1 billion.