SAP R/3 on DB2 for OS/390: DB2 Features That Benefit SAP

- DB2 SQL and system performance enhancements
- DB2 Utility performance enhancements
- Management tool enhancements

Viviane Anavi-Chaput
Steve Bergmann
Don Geissler

ibm.com/redbooks
SAP R/3 on DB2 for OS/390: DB2 Features That Benefit SAP

May 2001
First Edition (May 2001)

This edition applies to SAP R/3 for use with DB2 for OS/390 V5, V6, and V7.

This document created or updated on May 31, 2001.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2001. All rights reserved.
Note to U.S Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.
## Contents

**Preface** ............................................. v
The team that wrote this redpaper ............................. v
Special notice ........................................ vi
IBM trademarks ......................................... vi
Comments welcome ....................................... vi

**Chapter 1. Introducing DB2 features that benefit SAP R/3** .............. 1

**Chapter 2. SQL and system performance enhancements** ................ 3
2.1 SQL and system performance enhancements in DB2 V5 ............... 4
   2.1.1 Native ASCII data storage .................................. 4
   2.1.2 Dynamic statement caching .................................. 5
   2.1.3 ORDER BY clause (APAR PQ23778) .......................... 6
   2.1.4 DS MAX increased from 10000 to 32767 (APAR PQ18543) ....... 6
2.2 SQL and system performance enhancements in DB2 V6 ............... 8
   2.2.1 Index screening in RID list processing ....................... 8
   2.2.2 Unmatched column join for VARCHAR ........................ 9
   2.2.3 Outer join performance enhancements ....................... 10
   2.2.4 Uncorrelated subquery - indexable IN predicates .......... 10
   2.2.5 16 terabyte tables ........................................ 12
   2.2.6 255 tables per query/view ................................ 12
   2.2.7 Buffers and EDM pools in data spaces ...................... 13
   2.2.8 Alter Index to re-distribute partitions ................... 15
   2.2.9 Defer defining data sets (APAR PQ3099) ................... 15
   2.2.10 SET LOG SUSPEND/RESUME command (APAR PQ31492) ........ 16
   2.2.11 Access path selection adjustment (APAR PQ33429) ........... 17
2.3 SQL and system performance enhancements in DB2 V7 ............... 18
   2.3.1 Asynchronous INSERT preformatting ......................... 18
   2.3.2 UNION and UNION ALL operators ........................... 18
   2.3.3 DB2 Restart Light ....................................... 19
   2.3.4 Online system parameters ................................ 20
   2.3.5 Checkpoint frequency in minutes .......................... 21
   2.3.6 Long running UR warning .................................. 21

**Chapter 3. Utility performance enhancements** .......................... 23
3.1 Utility performance enhancements in DB2 V5 .......................... 24
   3.1.1 Online REORG ........................................ 24
   3.1.2 Data set reuse .......................................... 25
3.2 Utility performance enhancements in DB2 V6 ........................ 26
Preface

This IBM Redpaper describes the benefits that DB2 for OS/390 provides for SAP R/3. The paper discusses DB2 for OS/390 V5, V6, and V7 functions which address more specifically SAP R/3 data access, performance and maintenance requirements.

The team that wrote this Redpaper

This redpaper is produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Viviane Anavi-Chaput is a Senior IT Specialist for BI, ERP and DB2 at the IBM International Technical Support Organization, Poughkeepsie Center. She writes extensively, teaches worldwide, and presents at international conferences on all areas of Business Intelligence and DB2 for OS/390. Before joining the ITSO in 1999, Viviane was a Senior Data Management Consultant at IBM Europe, France. She was also an ITSO Specialist for DB2 at the San Jose Center from 1990 to 1994.

Steve Bergmann is an IBM IT Specialist in Availability Services. Steve consults with SAP R/3 for OS/390 customers in the areas of high availability, database backup and recovery, and disaster recovery.

Don Geissler is an IBM Consulting Technical Sales Specialist in ERP Solutions Sales of North America. He has been with IBM since 1983, providing systems engineering support to large S/390 customers. Don is a member of the ERP Competency Center S/390 team, working with SAP R/3 for DB2 for OS/390. His primary responsibilities are consulting with SAP R/3 for OS/390 customers in the areas of high availability, database backup and recovery, and disaster recovery.

Thanks to the following people for their contributions to this project:

Namik Hrle
SAP on IBM zSeries Software Development, Boeblingen, Germany

Thanks also to Terry Barthel and Al Schwab for their editorial assistance.
Special notice

This publication is intended to help technical professionals who work with SAP R/3 on DB2 for OS/390. The information in this publication is not intended as the specification of any programming interfaces that are provided by DB2 for OS/390. See the PUBLICATIONS section of the IBM Programming Announcement for DB2 for OS/390 for more information about what publications are considered to be product documentation.

IBM trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

IBM®

Comments welcome

Your comments are important to us!

We want our redpapers to be as helpful as possible. Send us your comments about this redpaper or other Redbooks in one of the following ways:

► Use the online Contact us review redbook form found at:
  ibm.com/redbooks

► Send your comments in an Internet note to:
  redbook@us.ibm.com

► Mail your comments to the address on page ii.
Introducing DB2 features that benefit SAP R/3

IBM and SAP have a long history of cooperation in supporting the large array of enterprises that use SAP R/3 with DB2 for OS/390 database server today—a solid partnership made possible by teamwork for development, integration, performance analysis, and customer service, to achieve excellent SAP R/3 results on the OS/390 platform.

The OS/390 platform offers unmatched scalability, availability, performance, and manageability to reliably handle any information needed in an SAP R/3 solution using DB2 for OS/390 as the database server.

The following sections of this paper discuss more specifically the DB2 for OS/390 feature enhancements that benefit SAP R/3:

- SQL and system performance enhancements in DB2 V5, V6, and V7
- Utility enhancements in DB2 V5, V6, and V7
- Management tool enhancements in DB2 V6, and V7
Chapter 2. SQL and system performance enhancements

This chapter describes DB2's SQL and system performance enhancements that benefit SAP R/3. The features are described for DB2 for OS/390 V5, V6, and V7.

Some of these features have been added over time, via APARs/PTFs, since the initial release of the DB2 for OS/390 version. If the feature was added via an APAR, we provide the APAR number in the description of the feature.

We discuss the following sections in this chapter:
- SQL and system performance enhancements in DB2 V5
- SQL and system performance enhancements in DB2 V6
- SQL and system performance enhancements in DB2 V7
2.1 SQL and system performance enhancements in DB2 V5

This section lists DB2 for OS/390 V5 functions that are particularly interesting for SAP R/3.

You can find a detailed description of these functions in the redbooks hereafter:

- *DB2 for OS/390 Version 5 Performance Topics*, SG24-2213

The following are DB2 V6 features which have been retrofitted into DB2 V5 via APARs/PTFs. They are described in Section 2.2, “SQL and system performance enhancements in DB2 V6” on page 8:

- Index screening in RID list processing (APAR PQ15670)
- Unmatched column join for VARCHAR (APAR PQ22046, PQ24933)
- Outer join (partially retrofitted) (APAR PQ18710)
- Uncorrelated subquery - indexable IN predicates (APAR PQ23243)

The following is a list of the DB2 for OS/390 V5 SQL and system features which are used by SAP R/3:

- Native ASCII data storage
- Dynamic statement caching
- ORDER BY clause (APAR PQ23778)
- DSMAX increased from 10,000 to 32,767 (APAR PQ18543)

2.1.1 Native ASCII data storage

DB2 V5 introduces support for ASCII character data encoding. System/390 systems use the EBCDIC encoding scheme and on other systems the character strings are represented using the ASCII encoding scheme. Two of the differences between EBCDIC and ASCII are the characters used for padding and the collating sequence. These differences are unacceptable to ERP applications like SAP R/3. To deal with these differences you can use field procedures (FIELDPROC) in previous releases of DB2. However, the CPU time required to execute these procedures can be high. Storing tables in ASCII format eliminates any need for conversion. This is the option select by SAP R/3.

By storing tables in ASCII format SAP R/3 saves CPU time on host by avoiding host variable conversions and column field procedures. Field procedures used for collation or conversion of data can degrade performance by as much as 30% for some transactions. SAP R/3 also saves CPU time on clients by not having to convert retrieved data.
2.1.2 Dynamic statement caching

SAP R/3 applications use dynamic SQL statements. Processing of a dynamic SQL statement requires two steps: prepare, then execute. Prepare is expensive because it consists of parsing and syntax checking, catalog searches for tables and columns, authorization checking, access path optimization, and creation of the executable statement.

For cases in which the same dynamic SQL statement is issued repeatedly, the ability to cache the previously prepared statement can significantly reduce the cost of running those statements as they will not have to be prepared over and over again.

Dynamic statement caching is activated at two levels:

- The first level is global and permits the caching in the EDM pool of all the SQL dynamic statements submitted to DB2. This level is enabled by setting the CACHE DYNAMIC SQL field to yes in the installation panel DSNTIP4. The corresponding ZPARM parameter is CACHEDYN. No SQL changes are needed to benefit from this function.

- The second level applies to a package/plan and is set during the bind operation by using the KEEP_DYNAMIC(YES) option. The storage used by the kept statements is controlled by setting the MAX KEPT DYN STMTS field of installation panel DSNTIPE. This function implies programming modifications.

Prepared dynamic SQL statements are cached in the EDM pool, so you should consider verifying the size of your EDM pool. See EDM pool size calculation in Chapter 2 of DB2 for OS/390 V5 Installation Guide for an estimate of the size of each prepared statement.

SAP R/3 has modified its own code to maximize the benefits it can get from this function. An SQL statement must have a perfect match to be able to reuse the prepared statement in the global cache; SAP R/3 uses parameter markers (or host variables), therefore the cache hit ratio can be very high. Using parameter markers considerably improves the cache hit ratio.

Dynamic statement caching achieves very good savings for SAP R/3 batch processes where a limited number of SQL statements get executed many times.

Dynamic statement caching has been improved through the following APARs: PQ14391, PQ09750, PQ11392, PQ11569, PQ09392, PQ12701, PQ12727, PQ13987, PQ14531, PQ14505, PQ14132, PQ17905, PQ14893, PQ19667, PQ14941, PQ14870, PQ07701.
2.1.3 ORDER BY clause (APAR PQ23778)

DB2 required that all columns referenced in an ORDER BY clause must also be named in the SELECT list of a query.

DB2 V5 now allows you to specify columns in the ORDER BY clause that are not in the SELECT list, such as the following statement:

```
SELECT name FROM q.staff.systables ORDER BY dept, years
```

You cannot use this enhancement of ORDER BY in conjunction with UNION, UNION ALL, GROUP BY, DISTINCT, and column functions such as MAX, MIN and SUM.

This enhancement is important for SAP R/3, which generates SQL statements.

2.1.4 DS MAX increased from 10000 to 32767 (APAR PQ18543)

Support has been added to permit allocation of data sets in excess of the 10,000 data set limit. This change is made in conjunction with changes introduced in OS/390 Version 2 Release 6 and later releases. Together, they enable DB2 to dynamically allocate a much larger number of concurrent data sets.

Systems running on versions of OS/390 which are prior to V2.6, continue to be restricted to the 10,000 data set limit.

The OS/390 Version 2 Release 6 allocation has been changed to remove the architectural limit of 10,000 concurrent, dynamically allocated DB2 data sets. This change addresses a scalability concern of large DB2 systems. This DB2 check, DS MAX, changes from the prior OS/390 limit of 10,000 data sets to the maximum of 32,767 data sets. Note the following considerations:

**Allocating a high number of data sets**

With OS/390 Scheduler Work Area (SWA) below the line, 10,000 open data sets is not generally possible. SAP R/3 requires the SWA above the 16MB line because of the large number of data sets that make up the SAP R/3 database.

Having a large number of allocated data sets has virtual storage and performance implications.

DB2 stop time can take longer as there are lots of data sets to close. If you are not able to specify STC NODETAIL for SMF, there is also a performance concern at DB2 shutdown. Abending DB2 instead of waiting for DB2 stop completion is not a recommended practice.
Checking the Storage Allocation
Calculation and checking for storage below the 16 MB line is more likely to cause a DSNT436I warning or DSNT437I error message from the DSNTINST installation clist saying storage below the line exceeds 5 MB and 8 MB with larger values of DSMAX. The parameter and maximum are shown on installations panel DSNTIPC.

The increase of DSMAX has very limited value for SAP R/3. SAP does not recommend specifying a DSMAX larger than 6000 due to the impact on DBM1 virtual storage below the 16 MB line.
2.2 SQL and system performance enhancements in DB2 V6

Here we describe the DB2 V6 functions that are beneficial to SAP R/3 on OS/390. Many of those functions have been retrofitted into DB2 V5 by APARs/PTFs.

You can also find a very detailed explanation of those functions in the redbook *DB2 UDB for OS/390 Version 6 Performance Topics*, SG24-5351.

DB2 for OS/390 V6 SQL and system functions that address SAP R/3 requirements are:

- Index screening in RID list processing
- Unmatched column join for VARCHAR
- Outer join performance enhancements
- Uncorrelated subquery - indexable IN predicates
- 16 terabyte tables
- 255 tables per query/view
- Buffers and EDM pools in data spaces
- Alter Index to Re-distribute partitions
- Defer defining data sets (APAR PQ30999)
- SET LOG SUSPEND/RESUME command (APAR PQ31492)
- Access path selection adjustment (APAR PQ33429)

2.2.1 Index screening in RID list processing

Index screening predicates reference columns in the index, but are not part of the matching index columns. For example:

```
SELECT * FROM T WHERE C1 = 1 AND C3 > 4 AND C4 = 6;
```

With an index on T (C1,C2,C3), C3 > 4 is an index screening predicate. It can be applied when accessing the index, but it is not a matching index predicate like C1 = 1. The value of MATCHCOLS in the PLAN_TABLE is 1. C4 = 6 is not an index screening predicate. Any column in the index that is not one of the matching columns for the query will be considered for index screening.

Prior to this enhancement, index screening was not used when RID list processing was involved. RID list processing is used by:

- List prefetch for single table access and hybrid join
- Index ANDing and ORing during multiple index access

DB2 now allows index screening during RID list processing to filter out additional rows at index access time. These rows no longer need to be passed on by the data manager for data page access and evaluation of the predicate. The number of RIDs that require sorting also decreases by this enhancement because some
of the rows have already been “screened out” at index access time. The optimizer filter factor calculations take this enhancement into consideration during list prefetch, index ANDing and ORing, and it is therefore possible to see the optimizer favoring those access path more often to achieve better response time.

SAP R/3 benefits naturally from this enhancement every time an application performs a list prefetch, or an index ANDing or ORing.

### 2.2.2 Unmatched column join for VARCHAR

You want to join two tables, but the data types or the length of the join columns do not match. DB2 can perform a join in that case, but there is a performance impact. Because of the mismatch on data type or column length, the join predicate is considered Stage 2. This means all qualifying rows of the inner table are passed back to the relational data system (RDS) component of DB2 for evaluation of the join predicate. If the join columns have the same data type and column length, the Data Manager (DM) component can deal with the join predicate at Stage 1.

**Example:**

```sql
SELECT * FROM T1,T2
WHERE T1.VARCHAR10 = T2.VARCHAR5
```

Suppose there is an index on T1 (VARCHAR10). The `T1.VARCHAR10 = T2.VARCHAR5` predicate is considered Stage 2 (and the index on T1 cannot be used either) because the column length is not the same. The same is true when the join predicate is `T1.VARCHAR5 = T2.VARCHAR5`.

This problem is now partially fixed with this enhancement. In order for a predicate with unequal attributes to qualify as a Stage 1 (and also indexable) predicate, it has to meet the following criteria:

- The predicate is dealing with a string data type (CHAR, VARCHAR).
- The join predicate is an equal predicate.
- The predicate is a Boolean term.

These enhancements have been incorporated in DB2 for nested loop and merge scan joins. Hybrid join does not support this feature.

A problem that is not being addressed by this enhancement is the mismatch of attributes on local predicates. For example, the local predicate `CHARCOL5 = 'ABCDEF'` is still stage 2 if the column length is smaller than the length of the literal.
All character strings in SAP R/3 are VARCHAR therefore SAP R/3 benefits naturally from this enhancement.

2.2.3 Outer join performance enhancements

With the introduction of the outer join support in DB2 V4, it has become much easier to write SQL outer join statements. This has increased programming productivity. However, the usability of outer join was sometimes limited by some of the performance downsides of using outer join, since more data needs to be examined and more processing takes place.

DB2 V6 introduces a large number of outer join performance enhancements making outer join SQL statements perform very closely to a similar inner join statement.

In addition, the SQL syntax of the ON clause has been extended as well to allow you to “boldly” write SQL that you could not write before.

The following enhancements is a list of the outer join enhancements implemented in DB2 V6 for OS/390. For a more detailed discussion of these enhancements see the redbook mentioned at the beginning of this section.

- SQL ON clause extensions
- Outer join predicates classification
- Join simplification
- Removal of unnecessary work files
- Aggressive merging of views and table expressions
- Aggressive predicate evaluation
- Predicate transitive closure for outer join
- Join order permutation
- Parallel outer join

Note that, as stated, some but not all of these enhancements have been made available for DB2 V5 for OS/390 via APAR PQ18710. The benefits for SAP R/3 are in significantly improved performance for a large number of outer join queries, especially in DB2 V6.

2.2.4 Uncorrelated subquery - indexable IN predicates

Before this enhancement, DB2 does not use a matching index when evaluating the IN predicate against the result set of a non-correlated subquery. The non-correlated IN subquery predicate is considered a stage 2 predicate. Take the following example:
**Example:**

```
UPDATE T1
SET SDATE = '01/01/1999', STIME = '20:38:35'
WHERE PROG IN (SELECT MASTER FROM T2 WHERE INCLUDE = 'TODAY');
```

- A unique clustering index exists on T1(PROG).
- An index exists on T2(INCLUDE, MASTER).

DB2 resolves the non-correlated subselect using matching index only access, sorts and removes the duplicates, and puts the results in a workfile. The PROG IN (subselect) is then evaluated. DB2 can use a non-matching index scan on T1(PROG) or a table space scan to access T1, and look for a qualifying row in the workfile. DB2 cannot use a matching index to look up the values in the subselect because the predicate is stage 2.

The DB2 code has been enhanced to evaluate whether it is beneficial to process the non-correlated IN subquery in more or less the same way as DB2 handles IN list index access today. However, for non-correlated IN subqueries, DB2 fetches from a duplicate free sorted workfile and not a list of values coded by the user. The subquery is executed at cursor OPEN time. The matching with the outer query block is not done until FETCH time.

Going back to example at the beginning of this section, the subquery on T2 is still executed first, the results are sorted in PROG order, and the duplicates are removed and stored in a workfile at OPEN cursor time. When the program starts FETCHing, instead of accessing the outer table T1, and matching the rows with the results of the subquery, DB2 now uses the values from the workfile to access the index on T1(PROG) using a matching index scan to evaluate the IN predicate. The resulting set of the non-correlated subselect is evaluated as the outer table in a “nested loop” way, while T1 is now considered to be the inner table.

The non-correlated IN subquery predicate has become indexable and stage 1. The DB2 optimizer evaluates if this transformation helps performance based on the existence of an index on the column specified on the left hand side of the IN predicate and the selectivity on the IN subquery predicates. This enhancement can be used in SELECT, UPDATE and DELETE statements.

Figure 2-1 on page 12 refers to a query used in a performance measurement study to evaluate this enhancement. The results on that figure show dramatic improvement both in elapsed and CPU time.
SAP R/3 benefits from this enhancement because these constructs are often used in SAP R/3 applications. Without this DB2 enhancement a major performance degradation was detected. The only way to solve it was to rewrite the application logic which is ‘curing’ rather than ‘preventing’ the problem.

2.2.5 16 terabyte tables

Version 6 of DB2 for OS/390 greatly expands the capacity to store data in a single table space. As shown in Figure 2, DB2 increases the limit for storing data in a single table space to 16 terabytes. This limit is up from 1 terabyte in Version 5 and 64 gigabytes in prior releases. You can create tables that can be up to 16 terabytes in size, either in compressed or uncompressed format, assuming that sufficient disk space is available. Of particular interest to SAP R/3 is that each partition of a large table can be 4 gigabytes as opposed to 2 gigabytes in prior releases. This means that developing the partitioning key could be less complex in DB2 Version 6.

2.2.6 255 tables per query/view

In prior releases of DB2, the maximum number of base tables in a view was 15. In Version 6, the number of tables that a view can support is 255. You can also specify 255 tables in SELECT, UPDATE, INSERT, and DELETE statements.
Some SAP R/3 are reaching the limit of 15-tables join. In the past users developing ABAP programs had to be aware of the 15 table limit. Raising the table limit in an SQL statement from 15 to 255 tables will benefit SAP R/3.

2.2.7 Buffers and EDM pools in data spaces

The capacity of exploiting large processor resources to run large application workloads is one of the strengths of DB2. Prior to DB2 V6, you allocate a buffer pool in either the DBM1 address space (virtual pool) or in a hiperspace (hiperpool).

The use of hiperpools helps to relieve the 2 GB addressability limitation of MVS address spaces. DB2 hiperpools reside in expanded storage only and may not contain changed pages. The total size of all hiperpools cannot exceed 8 GB. DB2 V6 provides an option to define buffer and EDM pools in a data space. Like hiperspaces, data spaces are data only address spaces. That is, no program code can run in those areas.

Data spaces provide a foundation for DB2 to exploit real storage larger than the 2 GB limit when the 64-bit machine becomes available. Until the new processor is available, we recommend that customers consider hiperpools first, before turning to data spaces, when running out of virtual storage in the DBM1 address space.

Whether you are using buffer and/or EDM pools in data space, take the following into consideration:

- Moving a data page back and forth between the data space and the look-aside pool may result in extra CPU usage.
- When using data spaces, make sure they are completely backed by processor storage. You do not want to see any paging activity when having to get to the buffers in the data space.

Buffer pool in data space

Unlike hiperspace, I/O can be directly done against buffers in data space (page movement occurs between the central storage and expanded storage). DB2 can also put changed pages in a virtual pool that resides in a data space, while a page in a hiperpool must be unchanged or written to DASD before it is allowed to be moved into the hiperpool.

Each data space can accommodate almost 2 GB of buffers and any single buffer pool can span multiple data spaces. However, no more than one buffer pool can be in a single data space. The sum of all data space buffers cannot exceed 8 million pages. This limit is independent of the buffer size.
A hiperspace is addressable in 4 KB blocks; in other words, it is page addressable; a data space is byte addressable. You cannot put a primary buffer pool into both a hiperspool and data space.

You define a buffer pool in a data space using a new VTYPE keyword on the ALTER BUFFERPOOL command. The possible VTYPE parameter values are PRIMARY or DATASPACE, each representing the following:

- **VTYPE(PRIMARY):** Allocates a virtual buffer pool in the DBM1 address space. The sum of all DB2 virtual buffer pools cannot exceed 1.6 GB.
- **VTYPE(DATASPACE):** Allocates a buffer pool in a data space. However, you need 128 bytes of buffer control storage in the DBM1 address space for each data space buffer.

The benefits of using buffer pools in data spaces are to:

- Improve buffer pool hit ratio. Since you can store more pages in a data space, than in a virtual pool that resides in DBM1 address space, pages can stay longer in memory.
- Allow for more parallel processing to execute prefetch I/O streams for large queries.

The main reason to choose a data space to store your virtual buffer pools is to provide relief for virtual storage constraints in the DBM1 address space and to provide greater opportunities for caching very large table spaces or indexes. If you are currently using hiperspools for read-intensive workloads and have not reached any DB2 virtual storage limit, there is no immediate benefit to moving to data spaces until processors are available that address more than 2 GB of real memory.

**EDM Pool in data space**

You can choose to have the part of your EDM pool that contains cached dynamic statements in a data space. By moving these “skeletons” of prepared dynamic SQL statements to a data space, you reduce the storage you require in the DBM1 address space.

If you specify YES for CACHE DYNAMIC SQL, DB2 will calculate a default value for the EDMPOOL DATA SPACE SIZE, automatically enabling the usage of a data space for cached dynamic statements.

Because SAP R/3 is primarily using dynamic SQL and this enhancement does allow for larger EDM pools, it will positively impact customers using dynamic statement caching.
2.2.8 Alter Index to redistribute partitions

Prior to DB2 V6 for OS/390, a change to the partitioning index to shift data from one partition to another was a complex process. You needed to follow the steps outlined below:

1. Stop the table space and start it read-only
2. Unload the entire table space
3. Drop the table space
4. Redefine the table space with new limitkey definitions
5. Redefine any indexes and NPIs
6. Reload the unloaded data
7. Run RUNSTATS
8. Rebind affected plans and packages
9. Reissue necessary grants for authorization to the original objects
10. Redefine any synonyms, views, and referential constraints
11. Take new image copies for recoverability

This process was complex and resource intensive, especially if you only needed to modify a subset of the total number of partitions.

In DB2 V6 for OS/390, you now have the ability to alter indexes in order to change the partitioning key values. Access to data is only restricted when the ALTER INDEX command with the LIMITKEY parameter completes for the affected partition(s). There is a new restrictive state REORG pending (REORP) that prevents access to these partitions until you do a REORG.

This function is especially useful for SAP R/3 where the data distribution will vary over time. Altering the keyranges without having to stop the entire table space will improve availability compared to the previous requirement.

2.2.9 Defer defining data sets (APAR PQ3099)

This new support allows DB2 users to use the DEFINE NO option in the CREATE TABLESPACE and CREATE INDEX SQL statements to defer the creation of underlying VSAM datasets for the created DB2 table space or index space. The undefined table spaces or index spaces will still have a DB2 catalog entry, but are considered as empty when accessed by SELECT or FETCH operation. An existing SQLCODE +100 (sqlcode100) is returned to any application which attempts to perform a read-only operation. Once the pageset is marked with 'undefined' state in DB2 catalog (the SPACE column in SYSTABLEPART or SYSINDEXPART is set to -1), it is treated as empty data set until the very first write operation occurs either through SQL statements or certain DB2 Utilities (such as LOAD). At the first write, DB2 resets the 'undefined' status in the catalog and creates the underlying VSAM data sets to allow the write operation. The 'undefined' status stored in the DB2 catalog will not be modifiable by any DB2 ALTER command or
any other third party utilities. DBAs and application package providers should consider to use the DEFINE NO option if the DDL performance is critical. The DEFINE NO option provides better management relief on DD limits and data usabilities by deferring the VSAM DEFINE/OPEN until the very first write.

Deferring the definition of data sets is an enhancement that can be useful for customers who use only a subset of modules from the full suite of applications provided by SAP R/3 - for example, FI, CO, or SD. Currently, customers receive all application tables, regardless of which applications they are actually going to use. This install method allows customers to add SAP R/3 modules easily after the initial install. On the other hand, it is possible for customers to have hundreds of empty tables for applications they will not use. These tables are perfect candidates to be defined using defer define.

### 2.2.10 SET LOG SUSPEND/RESUME command (APAR PQ31492)

Users have requested a way of temporarily “freezing” updates to a DB2 subsystem while the logs and database can be copied (e.g. using Enterprise Storage Server FlashCopy or RVA SnapShot) for remote site recover or prior point in time recovery usage. This would allow them to recovery the DB2 subsystem to a point in time without having to experience an extended recovery outage, or without having to stop or quiesce the primary system.

Avoid using this function while long running units of recovery are active. DB2 restart time is lengthened by long running updates.

New options are added to the -SET LOG command to be able to SUSPEND and RESUME logging for a DB2 subsystem. When a SUSPEND request is issued, a system checkpoint will be taken (in a non-data sharing environment), any unwritten log buffers will be written to DASD, the BSDS will be updated with the high written RBA, and the log-write latch is obtained to prevent any further log records from being created. This will prevent any further updates to the database until update activity is resumed with a -SET LOG RESUME request. The scope for these commands are single subsystem only, therefore the commands will have to be entered for each member when running in a data sharing environment.

For further details on how to use this command in conjunction with SAP R/3 database backup and recovery, see SAP note 83000 on the SAP Service Marketplace Web site.
2.2.11 Access path selection adjustment (APAR PQ33429)

DB2 V6 for OS/390 introduced a new parameter, NPGTHRSH, which will cause the DB2 Optimizer to favor index access for tables whose statistics indicate less than a given number of pages. For a given table, if NPAGES is less than the NPGTHRSH value, index access for the table will be preferred over a tablespace scan. After the initial install of SAP R/3, there are many empty or small tables which could grow rapidly in size. There are also a number of tables in SAP R/3 which are very volatile, meaning the number of rows can change very quickly and in large amounts. If a RUNSTATS is run on these tables when they are small, the DB2 optimizer would favor a tablespace scan, which would be inappropriate when the table grows.

Prior to this enhancement SAP R/3 recommended that the database administrator run a catalog update changing the statistics for small tables. Now SAP R/3 recommends that the customer use the new DB2 V6 parameter set as NPGTHRSH = 10. See SAP note 192320 for more detail; you can view it on the SAP Service Marketplace Web site (http://service.sap.com).
2.3 SQL and system performance enhancements in DB2 V7

This section describes the DB2 for OS/390 V7 functions that are beneficial to SAP R/3.

You can find detailed explanations of those functions in:
- DB2 UDB Server for OS/390 and z/OS Version 7 - Presentation Guide, SG24-6121
- DB2 UDB for OS/390 and z/OS release Planning Guide Version 7, SC26-9943

The DB2 for OS/390 V7 SQL and system functions that address SAP R/3 requirements are:
- Asynchronous INSERT preformatting
- UNION and UNION ALL operators
- DB2 Restart Light
- Online system parameters
- Checkpoint frequency in minutes
- Long-running UR warning

2.3.1 Asynchronous INSERT preformatting

DB2 improves the performance of insert operations by asynchronously preformatting allocated but unformatted data pages. When a new page is used for an insert, that page is close to the end of the formatted pages, and allocated but unformatted space is available in the data set—DB2 preformats the next range of pages.

With preformatting, an insert waits less often for a page to be formatted. When the preformatted space is used and DB2 needs to extend the table space, normal data set extending and preformatting occurs.

2.3.2 UNION and UNION ALL operators

The scope in which UNION and UNION ALL operators can be specified has been expanded. The CREATE VIEW statement, the INSERT statement, the UPDATE statement, the DECLARE GLOBAL TEMPORARY TABLE, nested table expressions in a FROM clause, and the subquery predicate are changed to allow a fullselect where a subselect was used in previous releases.
Now, you create a view by using UNION or UNION ALL to view the data as if it were in one table. If you use UNION ALL to combine the tables, the result consists of all the rows in the tables. If you use UNION, the result is the set of all the rows in the tables without duplicate rows. Whenever possible, the following optimizations are applied to the queries referencing such views, table expressions, or subqueries:

- The joins in the queries are distributed to the subselects of the UNION ALL.
- The query predicates are distributed to the subselects of the UNION ALL.
- Aggregation in the queries is distributed to the subselects of UNION ALL.
- Subselects that are not needed to answer the queries are eliminated.

The following example illustrates creation of a view that is the UNION ALL of three fullselects, one for each month of the first quarter of 2000. The common names for the views are SNO, CHARGES, and DATE.

```sql
CREATE VIEW DSN8710.FIRSTQTR (SNO, CHARGES, DATE) AS
    SELECT SNO, CHARGES, DATE
    FROM MONTH1
    WHERE DATE BETWEEN '01/01/2000' AND '01/31/2000'
    UNION ALL
    SELECT SNO, CHARGES, DATE
    FROM MONTH2
    WHERE DATE BETWEEN '02/01/2000' AND '02/29/2000'
    UNION ALL
    SELECT SNO, CHARGES, DATE
    FROM MONTH3
    WHERE DATE BETWEEN '03/01/2000' AND '03/31/2000';
```

You can use the INSERT statement in the same way you use fullselects. The UPDATE statement is also changed to support row-fullselect and scalar-fullselect, where row-select and scalar-subselect were previously supported in the SET assignment clause. In the DECLARE GLOBAL TEMPORARY TABLE statement, AS (subselect) DEFINITION ONLY is changed to AS (fullselect) DEFINITION ONLY. You now can use fullselect with a basic predicate, quantified predicate, EXISTS predicate, and IN predicate.

SAP BW exploits UNION in views.

### 2.3.3 DB2 Restart Light

In data sharing environments, the new LIGHT(YES) parameter of the START DB2 command lets you restart a DB2 member in light mode. Restart-light mode means that a DB2 data sharing member restarts with a minimal storage footprint and then terminates normally after DB2 frees retained locks.
Restart-light mode is intended for a cross-system restart in the event of an MVS system failure. The reduced storage requirement makes it possible to temporarily restart a DB2 data sharing member on a system that might not have enough resources to start and stop DB2 in normal mode. Releasing the locks with a minimum of disruption promotes faster recovery and data availability. For example, applications that are running on other DB2 members have quicker access to the data for which the failed member held incompatible locks.

You can also use restart-light mode in conjunction with the MVS Automatic Restart Manager (ARM). To have a DB2 data sharing member automatically restarted in light mode when system failure occurs, you must have an appropriately coded ARM policy. ARM does not restart the DB2 member again after a light restart is performed; the member terminates normally for the light restart.

2.3.4 Online system parameters

In SAP environments utilizing DB2 in a 24x7x52 mode, the need has been growing for online update of the major DB2 system parameters.

With DB2 V7 the new -SET SYSPARM command is introduced to dynamically reload the DSNZPARM load module. All parameters of the DSN6ARVP macro can be changed, and a large number of parameters from the DSN6SYSP and DSN6SPRM macros can be changed as well; see Figure 2-2.

<table>
<thead>
<tr>
<th>ZPARM</th>
<th>SAP Recommended Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTHREAD</td>
<td>Site specific</td>
</tr>
<tr>
<td>EDMPOOL</td>
<td>60000</td>
</tr>
<tr>
<td>EDMDSpac</td>
<td>100000</td>
</tr>
<tr>
<td>EDMBFIT</td>
<td>No</td>
</tr>
<tr>
<td>MAXRBLK</td>
<td>100000</td>
</tr>
<tr>
<td>CONTOBLK</td>
<td>Yes</td>
</tr>
<tr>
<td>DSMAX</td>
<td>6000</td>
</tr>
<tr>
<td>PCLOSEN</td>
<td>5</td>
</tr>
<tr>
<td>PCLOSET</td>
<td>10 min</td>
</tr>
<tr>
<td>DSSTIME</td>
<td>5 min</td>
</tr>
<tr>
<td>STTIME</td>
<td>30 min</td>
</tr>
<tr>
<td>SYNCAVL</td>
<td>30 min</td>
</tr>
<tr>
<td>STARJOIN</td>
<td>No (except for BW)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ZPARM</th>
<th>SAP Recommended Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHKFREQ</td>
<td>10 - 15 min</td>
</tr>
<tr>
<td>URECHKTH</td>
<td>1</td>
</tr>
<tr>
<td>URLGWTH</td>
<td>100K</td>
</tr>
<tr>
<td>Timeout</td>
<td>600 sec</td>
</tr>
<tr>
<td>UTIMOUT</td>
<td>3</td>
</tr>
<tr>
<td>RETLWAT</td>
<td>1</td>
</tr>
<tr>
<td>NUMLKUS</td>
<td>0</td>
</tr>
<tr>
<td>RELCURHL</td>
<td>Yes</td>
</tr>
<tr>
<td>SEQCAHC</td>
<td>Seq</td>
</tr>
<tr>
<td>SEQRES</td>
<td>No</td>
</tr>
<tr>
<td>CDSSRDEF</td>
<td>1</td>
</tr>
<tr>
<td>PARAMDEG</td>
<td>number of CPUs</td>
</tr>
</tbody>
</table>

Figure 2-2  DB2 subsystem online parameters of special interest to SAP systems
2.3.5 Checkpoint frequency in minutes

With DB2 V7, the checkpoint frequency parameter is enhanced to allow you to specify a range of minutes instead of a number of log records. Both options are available at install time and can be changed dynamically via commands.

This feature is useful in environments where the logging rate varies. You can maximize system performance by specifying the checkpoint frequency in time to avoid the performance degradation due to many system checkpoints taken in a very short period of time because of high logging rate. We recommend you set CHECKFREQ to a value between 10 and 15.

2.3.6 Long-running UR warning

Prior to DB2 V7, the warning for long-running unit of recovery (UR) was based on the number of checkpoint cycles to complete before DB2 issues a warning message for an uncommitted UR. But the number of checkpoints depends on several factors which may not include the long-running job.

With DB2 V7, the warning mechanism is additionally based on the number of log records written by an uncommitted UR. The purpose of this enhancement is to provide notification of a long-running UR that may result in a lengthy DB2 restart or a lengthy recovery situation for critical tables. The warning message is repeated each additional time the threshold is reached.

The value for written log records in the message is cumulative and indicates the number of log records written since the beginning of the UR. If statistics trace class 3 is active, an instrumentation facility component identifier (ICFID) 0313 is also written.

The UR log write check threshold is set in the DB2 parameter load module DSNZPARM (DSN6SYSP URLGWTH) at install time. The value may be modified using the -SET SYSPARM command. We recommend you use URCHKTH = 1 and URLGWTH = 100.
Utility performance enhancements

This chapter describes DB2 utility performance enhancements that are particularly beneficial to SAP R/3. The features are described for DB2 for OS/390 V5, V6, and V7.

Some of these features have been added over time, via APARs/PTFs, since the initial release of the DB2 for OS/390 version. If the feature was added via an APAR, we provide the APAR number in the description of the feature.

We discuss the following sections in this chapter:

- DB2 utility performance enhancements in DB2 V5
- DB2 utility performance enhancements in DB2 V6
- DB2 utility performance enhancements in DB2 V7.
3.1 Utility performance enhancements in DB2 V5

In DB2 V5, the ability to do online REORGs and data set reuse have been beneficial to users of SAP with DB2 for OS/390.

The DB2 for OS/390 V5 utility functions that address SAP R/3 requirements are:

- Online REORG
- Data set reuse

3.1.1 Online REORG

Online REORG significantly reduces the impact of a planned outage. The online REORG utility copies the tablespace to one or more shadow data sets where the actual reorganization is done. The DB2 log is applied in an iterative fashion to the shadow copy of the data to synchronize it with the actual online tablespace. After the log is applied, the shadow data replaces the unorganized data when the REORG utility changes data set names. The outage is now limited to read-only during the final application of the DB2 log to the shadow copy of the tablespace, and no readers or writers are allowed during the changing of the shadow data set names to the active tablespace data set names.

SAP recommends that the execution of the online REORG utility be limited to periods when the dialog processing is as low as possible and there is no SAP batch running. If limiting batch is not possible, ensure that the batch program commits every hundred records. Also consider isolating those tables that need to be reorganized online to a single table tablespace. This will minimize the impact to other SAP modules. The recommended settings for the online REORG utility are as follows:

- **SHRLEVEL CHANGE**
  Applications can read and write from the table that is being reorganized.

- **MAXRO 10**
  Specifies the amount of time for the last iteration of log processing. During that iteration, applications have read-only access.

- **TIMEOUT TERM**
  If the REORG utility gets a timeout condition while attempting to drain objects in either the LOG or SWITCH phases, then DB2 issues an implicit TERM UTILITY command, causing the utility to end with a return code 8 and leave the object in a RW state.

- **DRAIN ALL**
  DB2 will drain all readers and writers during the log switch phase
3.1.2 Data set reuse

Deleting and defining datasets within SAP, especially for the empty datasets, can slow performance and make it difficult to manage the placement of data sets. For storage group managed data sets, the option of reusing the data set can improve performance and data set management. The new keyword REUSE, is a new option for the LOAD, REBUILD, RECOVER and REORG utilities. REUSE indicates that table spaces and index spaces will be logically reset without a delete and redefine of the data set. Given that many tablespaces and indexes in an SAP environment can be empty, this change can have a significant impact on elapsed recovery times, particularly when it is necessary to do a point-in-time recovery, possibly recovering all DB2 objects in the SAP system.
3.2 Utility performance enhancements in DB2 V6

DB2 for OS/390 Version 6 offers significant enhancements to the DB2 utilities which benefit the SAP user. Many of these have been retrofitted to DB2 V5 by means of an APAR. We note these by including the APAR number in the list.

The use of these utilities is described in detail in Chapter 9 of *DB2 Server for OS/390 Version 5 Recent Enhancements - Reference Guide*, SG24-5421.

The DB2 for OS/390 V6 utility functions that address SAP R/3 requirements are:

- COPY and RECOVER
- LOAD and REORG
- REBUILD Index
- Fast Log Apply
- Inline RUNSTATS
- COPY with CHECKPAGE option (APARPQ25084)
- REORG enhancements (APAR PQ19077)
- Backup and recovery of indexes using image copies

### 3.2.1 COPY and RECOVER

To reduce the elapsed time of both COPY and RECOVER, DB2 Version 6 allows these utilities to process in parallel. You may COPY a set of DB2 objects (tablespaces or indexes) in parallel. During RECOVER, the tablespaces and indexes may be restored in parallel. In previous releases of DB2, the creation (during COPY) and the restoration (during RECOVER) of image copies was serialized. Full parallelism can be maintained provided that the copies are made to DASD. (Image copies to tape suspend the parallelism until the tape copy is created or restored.)

To fully exploit this feature, the DB2/SAP user has to take image copies to DASD.

### 3.2.2 LOAD and REORG

In DB2 Version 6, the LOAD and REORG utilities can build indexes in parallel. The DB2/SAP user will exploit the parallel index build feature of REORG (SAP does not explicitly invoke the DB2 LOAD utility).

### 3.2.3 REBUILD Index

With support for the recovery of indexes from image copies in DB2 Version 6, it is not likely that you will use the REBUILD index utility. However, if you find it necessary to use REBUILD to recreate indexes, the recreating or rebuilding of those indexes are done in parallel in DB2 Version 6.
3.2.4 Fast Log Apply

Prior to Version 6 of DB2, the log apply phase of RECOVER (for the RECOVER utility or recovery at restart time), was a serial process. The log record was read, the applicable tablespace page was read, the recovery at the data page level was effected; and the process was repeated for the subsequent log records.

With Fast Log Apply in DB2 Version 6, the log records are read and sorted by time of update within page within dataset. Tablespace pages are read only once using list prefetch. Parallel tasks are dispatched at the tablespace (or dataset) level. This significantly improves performance for the DB2/SAP user during both recovery and normal DB2 restart.

Fast Log Apply is also invoked during the Start Database command for LPL (Logical Page List) recovery and GRECP (Group Buffer Pool Recovery). This is particularly beneficial to the DB2 data sharing user.

SAP R/3 recommends that the Fast Log Apply buffer be set to 10 MB for normal operation and to 100 MB when performing a prior point in time recovery of the database.

3.2.5 Inline RUNSTATS

During the DB2 LOAD, REORG, and REBUILD utilities, inline statistics may be captured. This provides RUNSTATS statistics without a separate invocation of RUNSTATS and without reading the tablespace data a second time. DB2/SAP users will find this feature particularly beneficial in the REORG utility.

3.2.6 Copy with CHECKPAGE option (APAR PQ25084)

By specifying the CHECKPAGE option on the COPY utility DB2 will invoke routines that will perform checks on data and index pages. This will help ensure that the backup being taken is without errors.

3.2.7 REORG enhancements (APAR PQ19077)

With the REORG utility in DB2 Version 6, you can select rows to be discarded during REORG. The discarded rows may be written optionally to a discard file.

REORG UNLOAD EXTERNAL allows records to be converted to an external format for processing by a user-written application program. A LOAD utility statement is generated to allow discarded or unloaded rows to be loaded into another tablespace.
REORG has made improvements that can reduce potential deadlocks which benefits both performance and availability. Additionally there have been improvements in the Display Utility command and in auto termination of online REORG (APARs PQ20032 and PQ18941)

You can create a full image copy data set (SHRLEVEL REFERENCE) during REORG TABLESPACE execution. The new copy is an inline copy. The advantage to using inline copy is that the tablespace is not left in COPY pending status regardless of which LOG option was specified for the utility. Thus, data availability is increased.

You can determine when to run REORG for non-LOB table spaces and indexes by using the OFFPOSLIMIT, INDREFLIMIT catalog query options. If you specify the REPORTONLY options, REORG will produce a report detailing if a REORG is recommended; A REORG is not performed.

3.2.8 Backup and recovery of indexes using image copies

The COPY utility is enhanced to enable you to take a full image copy or concurrent copy of an index. You can then use the RECOVER utility to recover the index by restoring this image copy and then applying the log, just as you have always done to recover a tablespace. The process of recovering the index instead of rebuilding it can be more efficient and faster.
3.3 Utility performance enhancements in DB2 V7

DB2 for OS/390 Version 7 offers significant enhancements to the utilities which benefit the SAP/DB2 user.

For more details refer to:
- DB2 UDB Server for OS/390 and z/OS Version 7 - Presentation Guide, SG24-6121
- DB2 UDB for OS/390 and z/OS release Planning Guide Version 7, SC26-9943

The DB2 for OS/390 V7 utility functions that address SAP R/3 requirements are:
- Online REORG
- Statistics history
- UNLOAD
- Parallel LOAD
- COPYTOCOPY

3.3.1 Online REORG

The enhancement to Online REORG is the acceleration provided to some of its phases. There are three phases of an online REORG during which the data is totally or partially not available: the LOG, the SWITCH, and the BUILD2 phases. In DB2 V7, the processing in two of these phases, SWITCH and BUILD2, is accelerated, making these phases shorter. Therefore, the availability of the data for the SQL applications is much higher. The improvements are:

- Fast SWITCH phase

In the SWITCH phase, all data sets involved in the online REORG are no longer renamed, as this can be very time-consuming because of the overhead involved with VSAM's access method services (AMS). Instead, the DB2 catalog and directory are updated to point to the shadow data sets rather than to the original data sets.

The improvement to the SWITCH phase is an internal DB2 change which requires no changes within SAP. This enhancement is important for SAP which would have several hundreds of data sets to rename in the SWITCH phase. With the AMS renaming technique, the elapsed time of the SWITCH phase would be too long, preventing the SAP applications to access the data during that time, and making them timeout because of the wait time.
BUILD2 phase parallelism

When only some partitions of a partitioned table space are subject to an online REORG, the shadow non-partitioning indexes (NPIs) do not contain all index entries; hence, a replacement of the original NPIs by the shadow NPIs is not possible. Instead, the individual original index entries must be replaced by the shadow index entries. This processing is now done in parallel during the BUILD2 phase, thereby greatly decreasing the elapsed time of this step.

DRAIN and RETRY

The new parameters DRAIN and RETRY have been added to the REORG REFERENCE or CHANGE utility statement to control the time that the utility will wait to establish a drain, and also to enable you to retry waiting for a drain.

3.3.2 Statistics history

Statistics data can now be stored in the DB2 catalog on an historical basis that will enable us to do trend analysis. When RUNSTATS stores new information in the DB2 catalog tables, it will also store that information in the catalog history tables, so that the catalog will hold the current statistics, and the catalog history tables will hold historical (including current) statistics. The MODIFY STATISTICS command will be used to delete the outdated statistics in the history tables.

SAP is planning to use the statistics history enhancement.

3.3.3 UNLOAD

With the new UNLOAD utility, you can unload data from a table space or an image copy data set. In most cases, the UNLOAD utility is faster than the DSNTIAUL sample program and REORG UNLOAD EXTERNAL, especially when you activate partition parallelism. UNLOAD is easier to use and has more options than REORG UNLOAD EXTERNAL.

Figure 3-1 on page 31 shows how you can use UNLOAD to perform the following tasks:

- Unload data from an image copy data set
- Unload data from multiple partitions in parallel
- Select data by using a syntax similar to the SQL SELECT statement
- Select sample rows by table
- Use field selection, ordering, and formatting options
- Specify SHRLEVEL CHANGE or REFERENCE
3.3.4 Parallel LOAD

Using Version 7, you can easily load large amounts of data into partitioned table spaces. Parallel load with multiple inputs runs in a single step, rather than in different jobs.

The LOAD utility loads each partition from a separate data set so that one job can load multiple partitions in parallel. Parallel loading reduces the elapsed time for loading the data, as compared to loading the same data with a single job in earlier releases. If a table space has non-partitioned indexes (NPIs), using parallel load with multiple inputs is faster than using separate jobs for each partition. Using load parallelism is much easier than creating multiple load jobs for individual parts.

Figure 3-2 on page 32 shows a parallel load of four partitions, with the SORTKEYS keyword enabling a parallel index build of three indexes. Each load task takes input from a sequential data set and loads the data into a corresponding partition. The utility then extracts index keys and passes them in parallel to the sort task that is responsible for sorting the keys for that index. If the amount of data is too large to perform the sort in memory, the sort product writes
the keys to the sort work data sets. The sort tasks pass the sorted keys to their corresponding build task, each of which builds one index. If the utility encounters errors during the load, DB2 writes error and error mapping information to the error and map data sets.

![Parallel LOAD with multiple inputs and parallel index build](image)

**Figure 3-2  Parallel LOAD with multiple inputs and parallel index build**

### 3.3.5 COPYTOCOPY

While the COPY, LOAD, and REORG utilities can make two local and two remote site backup copies of data, you might want to make a single copy and then clone that copy at a more convenient time. The COPYTOCOPY utility asynchronously makes up to three additional backup copies from an existing copy and registers the copies in the DB2 catalog for recovery purposes. You can use object wildcarding and dynamic allocation capabilities with the COPYTOCOPY utility.
Management tools performance enhancements

This chapter describes DB2 management tools performance enhancements that are particularly beneficial to SAP R/3. The features are described for DB2 for OS/390 V6, and V7.

We discuss the following sections in this chapter:

- DB2 utility performance enhancements in DB2 V6
- DB2 utility performance enhancements in DB2 V7
4.1 Management tool performance enhancements in DB2 V6

DB2 for OS/390 Version 6 offers significant enhancements to the management tools which benefit the SAP/DB2 user.

4.1.1 DB2 PM Data Collector

The DB2 PM API was officially announced on May 2, 2000. A new manual is provided for those who want to use the API to code against. Also, the *DB2 PM Installation & Customization Guide* was extended to support installation and setup of Data Collector. SAP R/3 4.6C uses this new API to gather DB2 usage information for the database performance monitor transaction, ST04, which gathers performance data from the DB2 database server and displays it using the SAP GUI.
4.2 Management tool performance enhancements in DB2 V7

DB2 for OS/390 Version 7 introduces new management tools which can benefit the SAP/DB2 user.

4.2.1 DB2 PM Data Collector

Data Collector functionality has been extended to support dynamic SQL statement cache, and buffer pool data set statistics. Additional new functions supported by the Data Collector will be Sysplex Monitoring and Online Batch Reporting. SAP plans to utilize these new features when they become available.

Sysplex Monitoring, which provides a sysplex-wide view of the system, will be of special interest to SAP R/3 installations that have implemented DB2 data sharing.
Special notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.
The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.