IBM SAP International Competence Center

Coca-Cola Bottling Co. Consolidated utilizes SAP technical upgrade project to migrate from Oracle to IBM DB2

"Running the SAP Unicode conversion and the database migration together produced a perfect result: the combination needed no additional downtime, and the project completed on time."

Tom DeJuneas
SAP Systems Manager
Coca-Cola Bottling Company Consolidated

"By choosing to implement DB2 compression right away, we have reduced the database size by around 40 per cent. This gives us faster backup and reduced storage costs, and makes the SAP technical upgrades easier and quicker."

Andrew Juarez
SAP Lead Basis
Coca-Cola Bottling Company Consolidated

Coca-Cola Bottling Co. Consolidated migrates from Oracle to IBM DB2 during SAP technical upgrade

About this paper

This technical brief describes the migration of an SAP® R/3® Enterprise (version 4.7) non-Unicode system on Oracle® Database 9i to a Unicode system with IBM DB2® version 9. To achieve its business objectives, Coca-Cola Bottling Co. Consolidated (CCBCC) faced a technical upgrade of its SAP R/3 Enterprise system to SAP ERP 6.0. Completing this upgrade would require the company either to upgrade its existing Oracle database and purchase extra Oracle licenses, or to move to a different database platform.

The CCBCC team decided it was time to derive more performance from the business-critical SAP applications, while driving down hardware and software costs. Instead of upgrading Oracle, the team decided to introduce IBM DB2. As part of the SAP upgrade project, CCBCC's R/3 system would require a conversion to Unicode. By combining the Unicode conversion and the database migration, the customer realized many technical and cost advantages – for example, exploiting time savings by sharing backup and test phases.

Customer Objectives

- *Drive down software and hardware costs for the existing SAP software*
- *Complete the conversion / migration project on time and under budget*
- *Utilize the database migration project to avoid overhead for the Unicode Conversion*
- *Gain better performance for the SAP applications*
- *Reduce storage needs*
- *Remove workload from the database administrators*

IBM Solution

- *IBM DB2 9.1 for Linux, Unix and Windows, DB2 Storage Optimization feature (Deep Compression)*
- *IBM Power Systems™ servers (model p5-560)*

Customer Benefits

- *Combining the database migration with the SAP Unicode Conversion saved time, money and caused essentially no overhead in effort for the DB migration.*
- *Initial migration results show that even after the Unicode conversion, DB2 delivers a reduction in storage needs of approximately 40 per cent, as a result of Deep Compression of the SAP R/3 Enterprise 4.7 system.*
- *The duration of manufacturing runs was reduced from 90 minutes to just 30 – an improvement of more than 65 per cent.*
- *The migration was completed under budget and ahead of schedule, with less than 26 hours of planned downtime – saving time and costs.*
- *The company has reduced overall licensing and maintenance costs by avoiding the purchase of additional Oracle licenses.*
- *DB2 is easier to administrate and requires less attention from the SAP Basis database administrator – contributing to reduced costs.*
- *The company predicts savings in the next five years of about US$750,000.*

Background, starting point and objectives

Coca-Cola Bottling Co. Consolidated (CCBCC) makes, sells and delivers sparkling and still beverages, primarily products of The Coca-Cola Company. CCBCC is the second-largest bottler of Coca-Cola products in the United States, operating in eleven states, primarily in the southeast. Founded in 1902, CCBCC enjoys net sales of more than $1.4 billion and is headquartered in Charlotte, North Carolina.

Leveraging synergies: SAP Unicode conversion and DB2 migration

Prior to the technical upgrade of the SAP landscape, CCBCC decided to perform both a Unicode conversion and a migration from the existing Oracle database platform to IBM DB2 with Deep Compression. These changes would eliminate the need to buy new Oracle licenses, and would thus reduce the total cost of ownership (TCO).

By switching on the DB2 Deep Compression feature during the migration, the company was able to reduce the size of the database by more than 40 per cent – which will result in faster backups and shorter runtimes for the upcoming SAP software upgrade.

In the meantime, before the SAP upgrade, CCBCC can benefit from the highly automated DB2 database administration, offering reduced cost of operation. DB2 version 9 includes features such as self-managing storage, self-tuning memory management (STMM), automatic reorganization, automatic runstats, real-time statistics and backup via the integrated FlashCopy® feature.

All database administration and monitoring tasks can be completed from within the SAP Database Administrator (DBA) Cockpit for DB2 – an easy-to-use management environment integrated into the SAP application environment.

Deploying Unicode as a future-proof solution

CCBCC decided to deploy Unicode because all new SAP product releases (from SAP NetWeaver 7.0 onwards) will be based on the Unicode standard. CCBCC wanted to be prepared for new SAP applications such as SAP NetWeaver Process Integration (SAP NW PI), which are already part of future implementation plans.

In technical terms, the requirements for a Unicode conversion are very similar to those of a database migration. In both scenarios the customer must perform an export and import of the database using the SAP program R3load.

The Unicode conversion itself is executed during the export phase of the migration. It is therefore very easy to direct the database towards a new target system without additional effort and downtime. Migrating to IBM DB2 in conjunction with an SAP software upgrade and/or Unicode conversion leverages an opportunity to avoid duplicating project tasks such as backup and testing, and keeps the cost of the migration as low as possible.

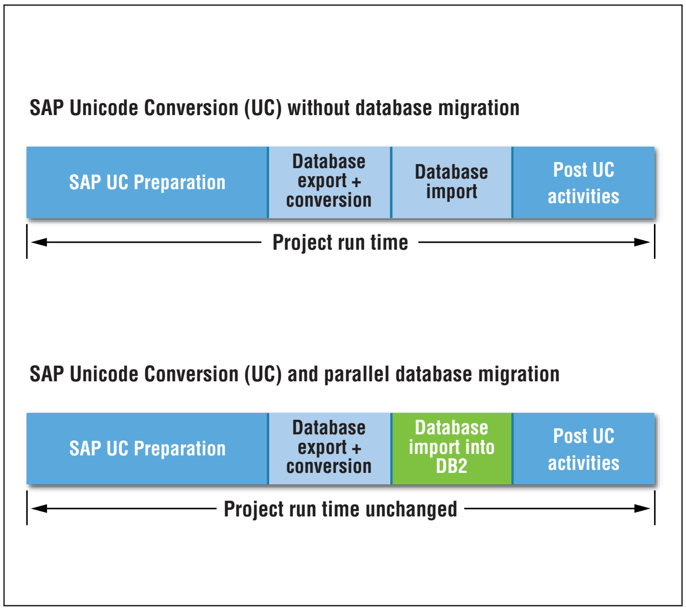

SAP Unicode Conversion (UC) without database migration
SAP UC Preparation
Database export + conversion
Database import
Post UC activities
Project run time
SAP Unicode Conversion (UC) and parallel database migration
SAP UC Preparation
Database export + conversion
Database import into DB2
Post UC activities
Project run time unchanged

Figure 1: Combined database migration with SAP Unicode (UC) conversion

Migration process – Heterogeneous System Copy

CCBCC used a standard SAP methodology for the migration process, known as the Heterogeneous System Copy (or OS/DB Migration) method. CCBCC was able to perform the migration and conversion during a scheduled maintenance window, so there was no need to make use of enhanced migration tools/ services from SAP such as Zero Downtime.

The migration project for the entire SAP R/3 Enterprise landscape took eight weeks in total, including two test iterations for the 1TB production database. The migration of the production SAP system itself was completed over one weekend, starting on the Saturday night and finishing in the early hours of Monday morning. The total downtime for the production migration was just 26 hours.

To achieve this reduced downtime, a set of SAP specific migration tools were used:

- Unsorted Export for the transparent tables
- Package Splitter for the largest tables (“big tables” group)
- Table Splitter for three large cluster tables
- Multiple instances of Migration Monitor to allow distributed parallel import and export processes
- R3load with Deep Compression option to activate compression during the migration phase.

The next part of this document depicts the way CCBCC utilized these tools, explains the reasons for the choices, and highlights the benefits.

Architectural overview – migration project at CCBCC

For the migration, CCBCC used four logical partitions (LPARs) on an IBM Power Systems server (model p5-560). Three LPARs were used to handle database export processes from the source system, and one LPAR was running the target system for the import processes. The export partitions consisted of a Central Instance / Database partition, which had 16 CPUs of 1.5GHz and 64GB of memory (CI/DB), and two other partitions that had four CPUs of 1.5GHz and 12GB of memory each. The import partition (or new CI/DB partition) had 16 CPUs of 1.5GHz and 64GB of memory.

During the testing phase, this system setup emerged as the optimal migration environment to handle the migration workload.

In order to meet the downtime objectives, the workload of the export packages were distributed between the CI/DB server and the other two servers (Hosts A and B) running in the first three LPARs. The CI/DB server handled the 3 largest cluster tables via Table Splitter. Host A handled the smaller tables. Host B was used to handle the export of the “big tables” group (which contained >10 million, >2 million, and >200,000 records); these were divided into smaller packages using Package Splitter. All three hosts used local storage to dump the export data to disk. Each export process was controlled by a Migration Monitor (MigMon) instance with its own configurations.

On the import side there was only one server – Host C (new CI/ DB server). The export disks of CI/DB, Host A and Host B were mounted via NFS (for reading) on Host C. The import was controlled by multiple MigMon instances.

From the “big tables” group on Host B, a subset was exported using the sorted unload option, which required additional CPU power and was one of the reasons for assigning an additional server for the export phase. During the import, the tables from the “big tables” group were compressed during the load process.

Database export – migration tools used

Unsorted vs. sorted export

CCBCC used both sorted and unsorted exports to unload the data from the Oracle database. In general, the unsorted export is faster than the sorted. But as CCBCC was also running a Unicode conversion, the migration team was forced to export the SAP cluster tables (for example CDCLS, RFGLG, EDI40) and SAP repository data classes via a sorted export. Sorting the data required additional CPU power, which was one of the reasons CCBCC handled the export phase with three servers.

- Sorted Export – Pool Tables, Cluster Tables, Reports, Dynpro's and Nametabs.
- Unsorted Export – Most of the transparent tables

With a sorted export, the pages of a table are read in the sequence of the primary key. If the cluster ratio is not optimal, data pages will not be read continuously. In addition, database sort operations may occur which will also extend the export runtime. By using the unsorted option, data is read sequentially and written directly to a file, instead of using an index that attempts to sort the data before writing to the file.

Unicode considerations for cluster tables

As a result of the Unicode conversion, the contents and the length of the records may change. Even the number of the physical records belonging to a logical record may change. Because the physical records are built together to form a logical record, the data must be read in a sorted manner to find all physical records that belong to a logical record. For these reasons, an unsorted unload is not possible.

Database limitations

DB2 supports unsorted exports, but some other databases only allow sorted exports. This represents a major roadblock in migrating away from these databases, and can also be a limitation in daily operations – for example, it is more difficult to set up test and QA systems using sorted exports. Especially for very large databases, being forced to run a sorted export will heavily extend the downtime window and make it almost impossible to change the database or even complete a Unicode conversion in a reasonable time.

Package and table splitting

The database size of nearly 1TB and the very large tables had been the determining factors for the downtime. CCBCC decided to parallelize the database export to improve the speed of the whole migration process, by using Package and Table Splitters.

Package Splitter splits tables of the source database into packages and exports them. In each case a dedicated R3load process handles each package. These processes can run in parallel and consequently make better usage of the CPU power. Table Splitter R3ta generates multiple WHERE conditions for a table, which are used to export the table data with multiple R3load processes running in parallel. Each R3load process requires a WHERE condition so that it can select a subset of the data in the table.

- 262 large tables ("big tables" group) were put in their own package using Package Splitter, to increase parallelism and ensure better granularity of the packages, resulting in better resource usage during the migration.
- 12 very large tables were divided into multiple packages using Table Splitter, enabling multiple R3load processes for parallel export and import of the table

- The remainder of the tables were combined in joint packages, using Package Splitter. By splitting the content to multiple R3load processes (20 parallel processes) it was possible to export and import the data in parallel, saving considerable time.

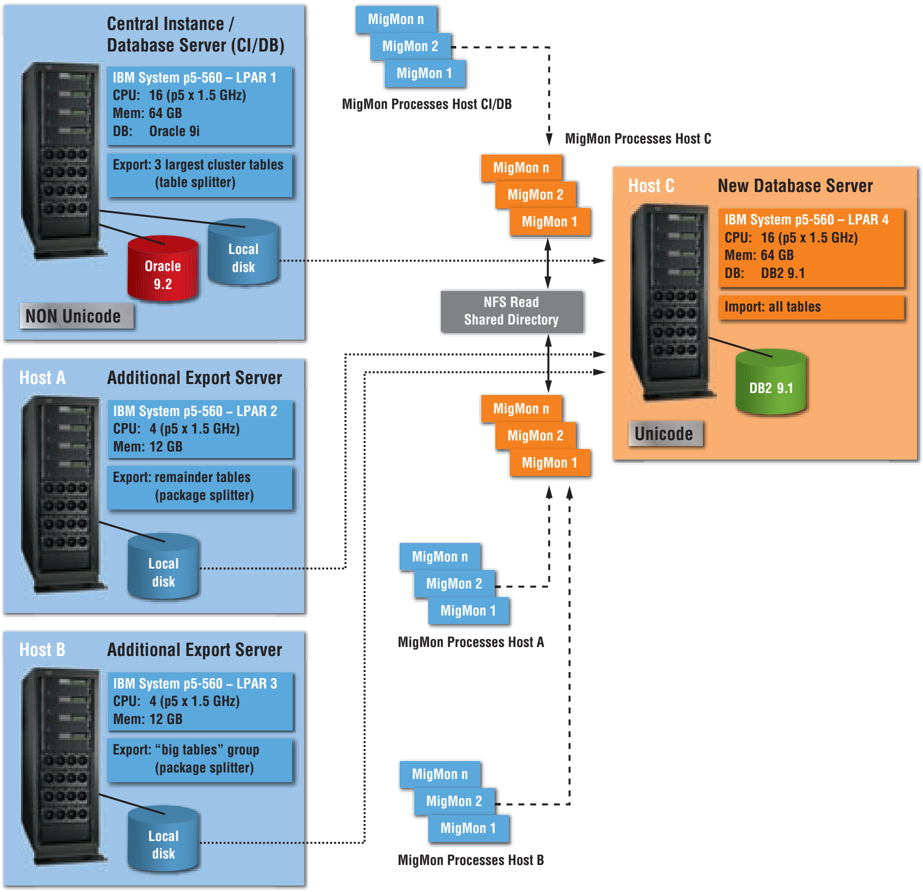


Figure 2: Landscape overview

Source:		**Target:**	
SAP release:	SAP R/3 Enterprise (version 4.7)	SAP release:	SAP R/3 Enterprise (version 4.7)
OS:	AIX 5.3	OS:	AIX 5.4
Database:	Oracle 9.2	Database:	DB2 9.1
Database size:	950GB	Database size:	575GB (with compression)
Data format:	Single codepage system without Unicode	Data format:	Unicode

Migration Monitor (MigMon)

In a Unicode conversion, the system copy causes very high CPU load during the export. Most of the CPU power is spent on data conversion, especially when processing cluster tables.

To avoid CPU bottlenecks, CCBCC distributed the exports and imports across 4 LPARs to parallelize these processes more effectively. This allowed CCBCC to take advantage of additional processor resources for the database export/import. The Migration Monitor helped to perform and control the unload and load process during the system copy procedure and enabled 20 export and import processes to be run in parallel.

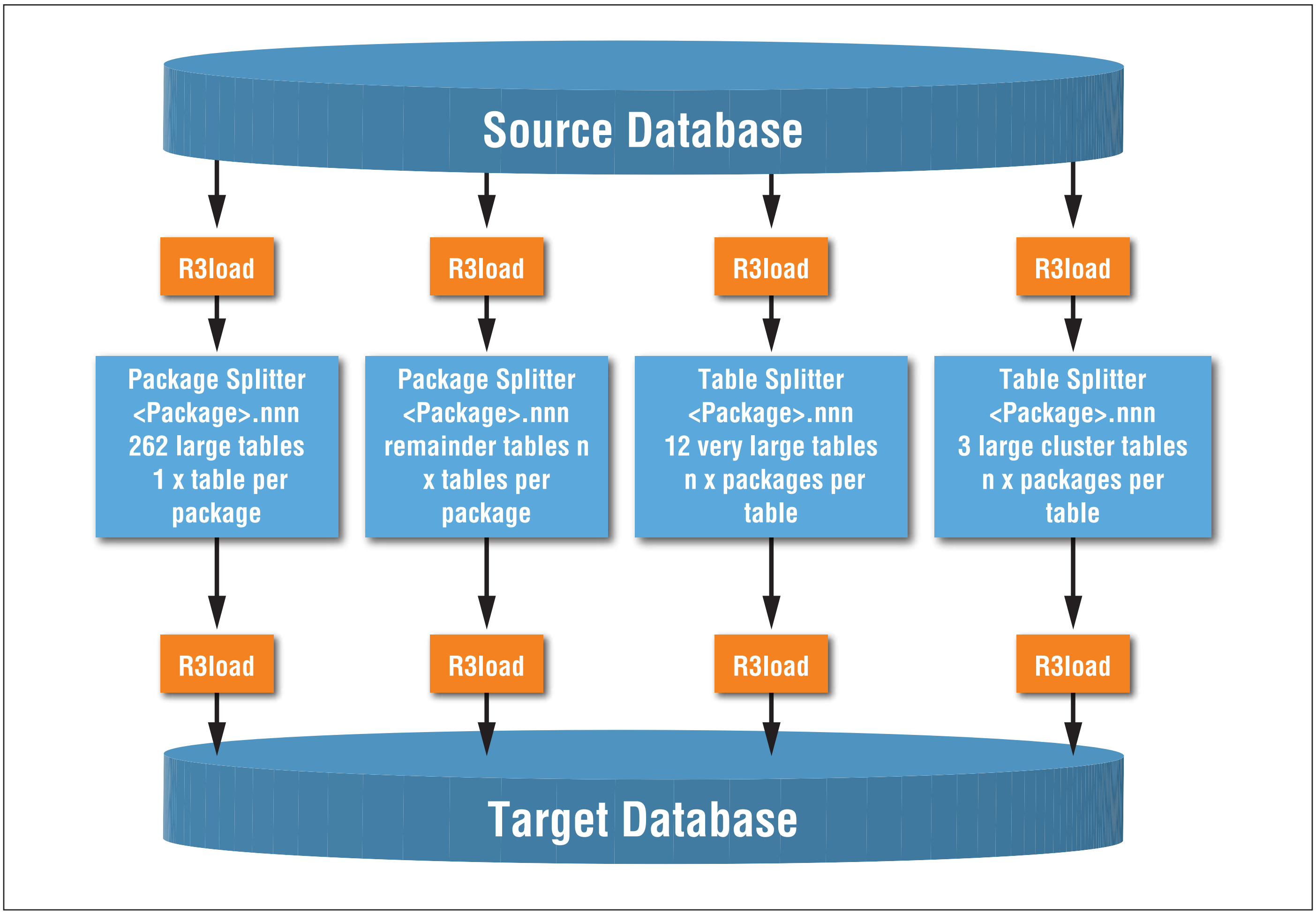

Source Database
R3load
R3load
R3load
R3load
Package Splitter <Package>.nnn 262 large tables 1 x table per package
Package Splitter <Package>.nnn remainder tables n x tables per package
Table Splitter <Package>.nnn 12 very large tables n x packages per table
Table Splitter <Package>.nnn 3 large cluster tables n x packages per table
R3load
R3load
R3load
R3load
Target Database

Figure 3: Package and Table splitting

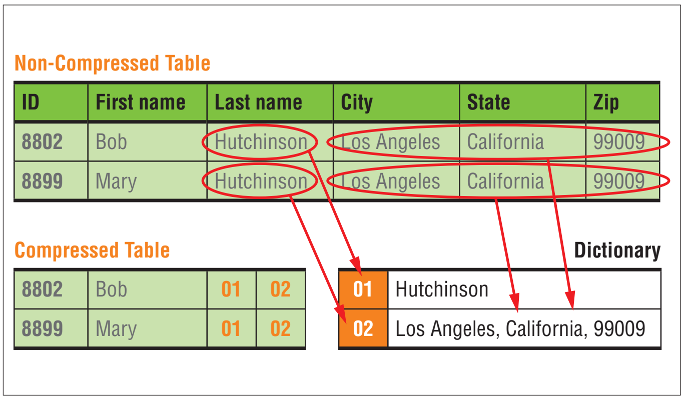
Database import - DB2 Deep Compression enabled

DB2 9 – Storage Optimization feature

The DB2 9 Storage Optimization feature – also called Deep Compression – uses a dictionary-based approach to replace repeating patterns with short symbols. The dictionary stores the patterns that occur most frequently, and indexes them with the corresponding symbols that are used to replace them. Due to the fact that all patterns within a table (not only within a single page) are replaced, impressive compression rates can be achieved (up to 90 per cent for single tables).

R3load with DB2 Deep Compression:

CCBCC wanted to make use of the benefits that the DB2 Storage Optimization feature offers right away, and decided to switch on Deep Compression during the migration process. Even with the knowledge that the compression rate with R3load version 6.40 might not be optimal, CCBCC decided to go ahead, and were rewarded with a compression rate of 40 percent and an impressive performance improvement. This was achieved despite the fact that only the 169 of the larger tables had been compressed.



Non-Compressed Table

ID	First name	Last name	City	State	Zip
8802	Bob	Hutchinson	Los Angeles	California	99009
8899	Mary	Hutchinson	Los Angeles	California	99009

Compressed Table

8802	Bob	01	02
8899	Mary	01	02

Dictionary

01	Hutchinson
02	Los Angeles, California, 99009

Figure 4: Deep Compression

Enabling DB2 Deep Compression during database migration and/or Unicode conversion is a very smooth way to compress the data at the time it is loaded into the database. The R3load tool provides several ways of deploying DB2 Deep Compression when the data is loaded into the tables. Depending on the version of R3load (i.e. version 6.40, or version 7.00 or higher), different options for compression are available, such as the new R3load 7.00 "SAMPLED" option.

This offers optimal data compression while avoiding time-consuming table reorganizations. In this paper we will focus on the compression feature of R3load version 6.40, as this was the tool used by CCBCC.

R3load 6.40 with compress option

To generate the compression dictionary, R3load first loads a defined number of rows into the table without compressing them. R3load creates the compression dictionary based on these rows by running an offline reorganization.

CCBCC incremented the value of the environment variable "DB6LOAD_COMPRESSION_THRESHOLD" to define the number of rows that would be initially loaded and used to create the dictionary. The default value for this threshold is 10,000 records, which was too low to provide optimal compression sampling for the larger tables.

By sampling between 10 and 80 percent of the records (depending on the number of rows in the tables), CCBCC was able to set optimal threshold values and achieve very good compression results. The two largest tables (COEP, BSIS) contained more than 130 million records, followed by several tables with between 10 and 70 million records.

CCBCC grouped the compressible transparent tables using the following row count thresholds:

- Group of 20 tables of more than 3 million records – threshold = 3 million
- Group of 47 tables of more than 200,000 records – threshold = 200,000
- Group of 102 tables of more than 60,000 records – threshold = 60,000

Note that not all tables matching the thresholds were flagged for compression and added to those groups. Only the ones which showed good compression results in the test phase were selected.

After the initial import and the creation of the dictionary, R3load imports the remaining rows into the table and DB2 compresses the data based on the dictionary.

Tables that are intended for compression during the load phase must have the compression attribute switched on. Since CCBCC had some tables that should be compressed and others that should not, different template files for the Migration Monitor were used.

CCBCC ran the import with several instances of the Migration Monitor (in different directories – see figure 2), and used different values for DB6LOAD_COMPRESSION_THRESHOLD for each instance.

Summary

Combining the Unicode upgrade with a database migration paid off for CCBCC – enabling the company to leverage synergies throughout the whole migration process, and eliminate the duplication of processes such as backup and testing. The whole ERP migration project took about eight weeks from start to finish, including the Unicode conversion.

Another essential aspect was the easy transfer of database management skills from Oracle to DB2, and the user-friendliness of DB2. CCBCC had strong in-house Oracle skills, and yet in a matter of weeks the database administrators became fully competent on DB2 – a tribute to the ease of transition to DB2 for experienced DBAs, regardless of their technical legacy.

CCBCC was able to benefit right away from the value DB2 offers:

- Lower TCO
- 40 per cent reduction in database size
- Better performance – manufacturing runs are over 65 per cent faster
- Better integration of the database in SAP tools (SAP DBA cockpit for DB2)
- Reduced DBA workload to manage and administrate DB2

With DB2 in place, CCBCC is well prepared for the upcoming upgrade to SAP ERP 6.0, which can now be performed much more smoothly and rapidly. The reduction in database size by 40 per cent will result in faster backup and shorter runtimes for the SAP software upgrade.

65
64
63
62
Coca-Cola
CLASSIC

61
60
59
Coca-Cola
CLASSIC

For more information:

To learn more about the solutions from IBM and SAP, visit: **ibm-sap.com**

For more information about SAP products and services, contact an SAP representative or visit: **sap.com**

For more information about IBM products and services, contact an IBM representative or visit: **ibm.com**

Contacts:

IBM

Elke Hartmann-Bakan
Global Information Management SAP
Technical Marketing/Sales Enablement
(elke.hartmann-bakan@de.ibm.com)

For further questions please contact the IBM SAP International Competency Center via **isicc@de.ibm.com**

IBM Deutschland GmbH
D-70548 Stuttgart
ibm.com

Produced in Germany

This brochure illustrates how IBM customers may be using IBM and/or IBM Business Partner technologies/services. Many factors have contributed to the results and benefits described. IBM does not guarantee comparable results. All information contained herein was provided by the featured customer/s and/or IBM Business Partner/s. IBM does not attest to its accuracy. All customer examples cited represent how some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication is for general guidance only. Photographs may show design models.