Redpaper

Paolo Bruni
Clauder Balzano

Converting Adabas to IBM DB2 for z/OS with ConsistADS

Consist Advanced Development Solution (ConsistADS) is an end-to-end conversion solution that combines conversion and transparency methods for migrating to IBM DB2 for z/OS software. The solution includes DB2 for z/OS and several DB2 tools as part of the package.

This IBM Redpaper publication explains the Natural and Adabas[1] conversion to DB2 for z/OS by using ConsistADS. It includes prerequisite technical assessment requirements and conversion challenges. It also describes a real customer conversion scenario that was provided by the IBM Business Partners that facilitated these conversions for customers.

This paper includes the following topics:

- Natural and Adabas conversions
- Adabas requirements
- ConsistADS automated conversion
- How ConsistADS maps Adabas data files to DB2
- ConsistADS conversion process

[1] Also written as ADABAS. It stands for *advanced database system*. Adabas is often used in conjunction with Software AG's programming language, Natural.

Natural and Adabas conversions

Companies used the Natural language and Adabas database to write mission-critical applications that are core to their business. However, Natural applications that are written in a pre-relational database, such as Adabas, are no longer mainstream and do not follow accepted IT industry standards. This shortfall makes Natural mature applications more difficult and expensive to maintain. Also, the applications cannot offer the greater flexibility of delivering new information services that relational databases do.

When customers start a conversion process from Natural and Adabas to DB2 for z/OS, they want to use the relational database capabilities and features that are offered by using relational database architecture. These capabilities and features provide a flexible database platform that meets the following business requirements:

- Web-enabled applications
- Service-oriented architecture and business process management (SOA and BPM)
- Data warehouse
- Business analytics
- Easy recoverability
- Two-phase commit
- Increased synergy with security
- Utilities performance
- SQL flexibility
- Uses tables larger than 4 billion rows
- Use of DB2 for z/OS features, such as partitioning and parallelism
- Data sharing
- Large objects
- Stored procedures
- Triggers
- Referential constraints

Adabas requirements

When we think about an Adabas conversion, it is possible to treat each Adabas file as a relational DB2 table. However, this simplistic assumption must not be made. Many Adabas database designs create multiple data type records within a single Adabas record, which is counter-intuitive to the relational table normalization philosophy. Because of the major differences in the data file system design models, a database administrator often is required to split Adabas files into several tables to reap the full benefit of a relational model design. A relational database system also allows the users of a client to query any table column by the name of the column based on a value, which is a capability that is not always allowed for in an Adabas file design.

The following steps are used to convert Adabas data to DB2 for z/OS:

1. Verify that each Adabas file is in third normal form. The file must be normalized if it is not already.
2. Create a DB2 table for each resulting file.
3. Translate each Adabas field to a DB2 column.
4. Identify a primary key for all defined tables.
5. If referential integrity is implemented, a foreign key must be defined for each dependent table.

You can decide between the following conversion and transparency solutions, depending on your specific needs:

- Conversion

  In this method, customers convert all languages and the database. They manually migrate their Natural online and batch programs to Java or COBOL, COBOL with IBM CICS, and Adabas to DB2. This conversion process is slower because there are many objects to manually convert (source code modules, data modeling, tables definitions, and so on) to create these objects in the new environment.

- Transparency

  In this method, customers convert only the Adabas database and use conversion programs to intercept the Adabas calls of the current Natural program and replace the calls with SQL calls to DB2 for z/OS. This conversion occurs when customers are reluctant to migrate their Natural code, so they choose to keep Natural running with the same source code as a new object that runs outside the run time of the Natural program. This method often lowers risks and speeds up the project. It is combined with the conversion solution that is completed later.

Table 1 compares the two methods.

Table 1 Conversion versus transparency

	Conversion	**Transparency**
Risk	Medium	Low
Project timing	Slower	Faster
DB2 use	Medium	Lower
Proprietary licenses	Not needed	Needed
Maintenance costs	Lower	Higher
Performance	Higher	Lower

Conversion challenges

The following challenges must be considered when a customer chooses an Adabas conversion method:

- Adabas inventory
- Disk storage estimates
- Data relationship
- Performance
- Applications

These challenges are reviewed next.

Adabas inventory

Estimating the workloads, costs, and schedules for the project is important. Therefore, it is necessary to have a thorough inventory of Adabas and application data volumes. Customers that have the PREDICT data dictionary generate reports that provide all of the requisite information.

A conversion project is an opportunity to assess what is required and to plan accordingly. Many programs serve only the development environment and are not used.

Disk storage estimates

Helping customers estimate the amount of disk space that is used in the DB2 for z/OS is important. In some instances, DB2 might require more disk space than Adabas.

Data relationship

The goal of the conversion is to avoid redundant data by normalizing the data model. The chosen conversion tool must provide flexibility to help the customer decide whether to denormalize the multiple occurring field (MU) and the periodic group (PE) field. However, normalization is often used to create new tables.

The tool must conduct the data type conversions and the character conversion according to the UNICODE, ASCII, and EBCDIC formats.

Adabas allows field names to include hyphens, as shown in Example 1. The tool must convert these field names to the accepted DB2 for z/OS standards.

Example 1 Adabas fields names

```
VIEW   : CUSTOMERS                          DEF.SEQ:   DBID:1   FNR:  45
COMMAND:
I T L DB NAME                          F LENG  S D       REMARK
- - - -- -----------bottom------------ - ----  - - ------------------
    1 CU CUSTOMER-NAME                 A 8.0     D
  G 1 SZ STATE-ZIP
    2 ST STATE-NAME                    A 2.0     D
```

The NULL value is treated differently between the databases. Adabas defines blank alphabetic fields and zero numeric fields as “null.” In DB2, null values are neither zero nor blank; they are unknown. A way to address this difference is to define the new columns by using NOT NULL WITH DEFAULT.

With alphanumeric fields, Adabas allows for storing a field that is longer than its definition; therefore, you use the VARCHAR data type to manage the field.

Performance

Before you start the process to convert to DB2 for z/OS, you must consider the following issues:

- Avoid too much normalization of the physical model, because that might cause DB2 for z/OS performance degradation.
- Apply filter factors to avoid reading unnecessary data.
- Keep the same indexes used in Adabas and improve the indexes whenever possible.
- Understand the DB2 sort process.
- Avoid the use of cursors when only one row is retrieved.
- Avoid the use of numerous joins because of the creation of Natural data definition modules (DDMs).
- Use CURSOR for UPDATE only in the necessary columns.

Applications

The conversion of the Natural procedural language is one of the most challenging steps for the migration. This conversion is challenging because of the manner in which Natural compiles all of the objects (MAP, PROGRAMS, SUBPROGRAM, DDM, and so on), and commands when accessing the Adabas database.

Suppose that the customer chooses to migrate Natural programs to COBOL, Java, or EGL language, but this scenario leads to manually rewriting many programs. Alternatively, a customer chooses to keep running Natural programs and access DB2 for z/OS through a transparency layer solution. The conversion is simpler by using COBOL programs that access Adabas because only the ADASQL calls must be changed to standard SQL statements to be pre-compiled. ConsistADS converts the programs to Advance, a fully compatible language to Natural.

In addition to the changes required to convert Adabas calls to DB2 SQL, there might be application logic changes required as a result of the DB2 design. The following application changes result from data changes made to resolve differences between an inverted list, nonrelational database, and a relational database:

- Normalization effect
- Record splitting
- Repeating groups
- Record to multiple tables
- Translating group level fields
- Perform **`READ LOGICAL`** by using DB2 cursors

In addition, the following tasks must be applied:

- Convert each Adabas (Natural) program to a corresponding DB2 program wherever possible.
- Determine the naming conventions.

ConsistADS automated conversion

There are many such solutions available to help customers who decide to convert their Natural applications and Adabas to DB2 for z/OS.

Consist Software Solutions Inc. (Consist) developed a solution that combines the benefits of the conversion and transparency methodologies. Consist examined numerous approaches to the conversion and designed a solution based on the following demands of the client:

- Faster conversion
- Lower risk
- Increased cost savings
- Automatic functionality enhancement of the applications

Conversions that are conducted by Consist are highly successful and the company is positioned as specialists in IBM DB2 for z/OS when a *low-cost* and *short conversion period is* required. Consist allows customers to convert multiple applications simultaneously.

The important decision for any customer is to choose a solution that replaces Natural and Adabas from a vendor capable of providing an end-to-end automated conversion solution. A vendor also must have extensive experience in working with Natural and Adabas, converting applications to a DB2 environment, and a working knowledge of application development in the IBM System z environment.

Consist offers ConsistADS to convert Natural and Adabas to DB2 for z/OS. Consist has decades of experience in delivering solutions that are written in Natural and Adabas in the System z environment.

ConsistADS is an end-to-end conversion solution that includes DB2 for z/OS and several DB2 Tools as part of a packaged solution that is delivered to the customer.

The ConsistADS solution minimizes the risk factor to and concerns of a customer during the conversion from Natural and Adabas to DB2 for z/OS by using the Consist automated conversion engine.

Combining methodologies: This solution uses a combination of conversion and transparency methodologies. The Adabas database is converted to DB2 for z/OS and the Natural language is converted to Advance, the language used by Consist. The Advance language runs inside the runtime environment that is provided by the solution.

ConsistADS overview

ConsistADS delivers complete automatic conversion and execution of business applications that are written in Natural and Adabas. ConsistADS with DB2 for z/OS offers a set of tools that automatically runs the conversion from Adabas database to DB2 and converts NATURAL programs, including mapping the original Adabas access calls to SQL commands.

ConsistADS monitor allows high availability, transparency, and SNA integration because it behaves as a transaction coordinator through XCF protocol in z/OS. Multiple instances of the ConsistADS server in z/OS or in DB2 for Linux, UNIX, and Microsoft Windows can be reached by its IBM VTAM interface, which is used to communicate with the SNA network to provide transparent operation with 3270 terminals and to participate in an IBM CICS transaction environment.

ConsistADS with the ConsistDMS component integrates DB2 for z/OS database and converts the Adabas hierarchical data file structures into DB2 SQL Data Manipulation Language (DML) objects.

Included in the ConsistADS solution is an integrated development environment (IDE) platform that is accessible through a web browser. By using the IDE, the client maintains existing converted applications and performs future application development to add new functions (such as business analytics modules) to the converted applications.

ConsistADS preserves the accumulated application development expertise of the client because ConsistADS enables a client to maintain converted and new applications in Advance, which is a language fully compatible with Natural original source code in the z/OS platform.

ConsistADS IDE combines functions that are used by any organization. ConsistADS IDE also allows the organization to develop business applications quickly and efficiently by taking advantage of the benefits of the latest information technology.

Editing, compiling, debugging, installing, and distributing the functions of the ConsistADS objects allows customers to develop and test an application in a z/OS controlled environment. This flexibility results in IT cost reductions and increased productivity by optimizing IBM System z resources.

Organizations often gain the following benefits when using ConsistADS:

- Preserves existing application investments
- Automatically converts applications from the original sources
- Converts Adabas hierarchical data into DB2 relational structures
- Provides a robust communication layer for efficient integration to optimize resources and maximize the performance of an application
- Provides thorough security by using ADS modules and IBM RACF software

- Replaces the use of TP monitors by delivering online web services directly
- Transparently provides automatic web functionality to applications
- Offers an IDE development platform with access through a web browser
- Allows the use of other development environments for editing and maintaining existing programs
- Controls the versions in distributed and collaborative environments
- Supports sysplex environment, ensures scalability, and supports large numbers of concurrent users
- Coordinates multiple instances of online servers either in z/OS or in DB2 for Linux, UNIX, and Windows systems
- Directly connects to SNA allowing use of 3270 devices
- Allows CICS integration for transaction and security control, and delivers a remote development option that allows for better use of resources and increased support for the IBM Mainframe© z/OS platform
- Offers optional Business Analytics integration in addition to components for implementing SOA and integrating BPM tools

ConsistADS solution

With ConsistADS suite, Natural and Adabas-based application systems are fully migrated and modernized. In addition to saving significant licensing and maintenance costs, customers benefit from DB2 for z/OS database and modern web User Interface technology.

ConsistADS is a complete IDE suite that features a compiler, runtime system, and migration toolset. The special feature of ConsistADS is the capability to automatically migrate and run multiple application systems that are coded in Natural in a single conversion step.

ConsistADS provides all of the tools necessary to preserve the original investment in mature applications that are written in Natural. The source code automatically is converted to Consist Advance, the 4GL language of Consist that is compatible with Natural. The full scope of the functionality of the language is run in ConsistADS with DB2 for z/OS.

ConsistADS is an IDE platform that includes the Advance language, the RDBMS, and the ConsistDMS components for the conversion of the Adabas data to an RDBMS, such as DB2 for z/OS. Ancillary DB2 tools are offered and packaged directly by Consist under a global OEM software partnership with IBM for the integration of IBM products into ConsistADS.

Customers experience the following benefits of the ConsistADS solution with IBM software:

DB2 for z/OS tools

Consist includes a DB2 for z/OS license in their ConsistADS solution and (at the request of the customer) important DB2 for z/OS tools. The DB2 tools are the part of the solution that provides the following system management tasks that are related to DB2 for z/OS:

- Application management
- Backup and recovery
- Business intelligence and dynamic warehousing
- Data governance
- Data replication
- Database administration and change management

- Performance management
- Utilities management
- Version upgrade acceleration

For more information, see the IBM DB2 tools for z/OS web page:

http://www.ibm.com/software/data/db2/zos/tools/

The following set of DB2 for z/OS tools is provided by Consist at the request of the customer:

- DB2 Administration Tool for z/OS

  This tool simplifies the complex tasks that are associated with safely managing DB2 objects and schemas throughout the application lifecycle with the least possible effect on availability.

- IBM Tivoli OMEGAMON XE for messaging for DB2 Performance Expert on z/OS

  This tool evaluates the efficiency and optimizes the performance of the DB2 for z/OS environment.

- DB2 Utilities Suite for z/OS

  The Utilities Suite minimizes the downtime that is associated with routine DB2 data maintenance and ensures the highest degree of data integrity.

- DB2 SQL Performance Analyzer for z/OS

  This tool helps improve DB2 application design to achieve maximum productivity by testing different “what if” scenarios.

- DB2 SQL Tuning for z/OS

  The DB2 SQL Tuning for z/OS tool enables efficient customization and tuning of SQL workloads and DB2 objects.

- DB2 Query Monitor for z/OS

  This tool enables the customer to efficiently customize and tune the SQL workload and DB2 objects.

ConsistADS is packaged with IBM DB2 for z/OS and related DB2 tool products in the ConsistADS DB2 for z/OS solution set.

The Consist solution must be installed in an IBM System z New Application License Charges (zNALC[2]) LPAR. Usage charges for DB2 and any DB2 utilities are based on the size of the zNALC LPAR.

The ConsistADS products

As Figure 1 on page 9 shows, the following tools are included in the ConsistADS solution architecture:

- Mainframe monitor
- Virtual machine (runtime system)
- Online server
- Common Web Architecture (CWA)
- Compiler for applications that are written in Natural or Advance
- Editing software for the web
- Data migration system

[2] zNALC extends the IBM commitment to subcapacity pricing, allowing customers with a Qualified Application to obtain a reduced price for z/OS. Charges are based on the size of the LPARs that are executing a Qualified Application, assuming that all applicable terms and conditions are met.

- Security application
- Integration infrastructure
- Web interface customization tool

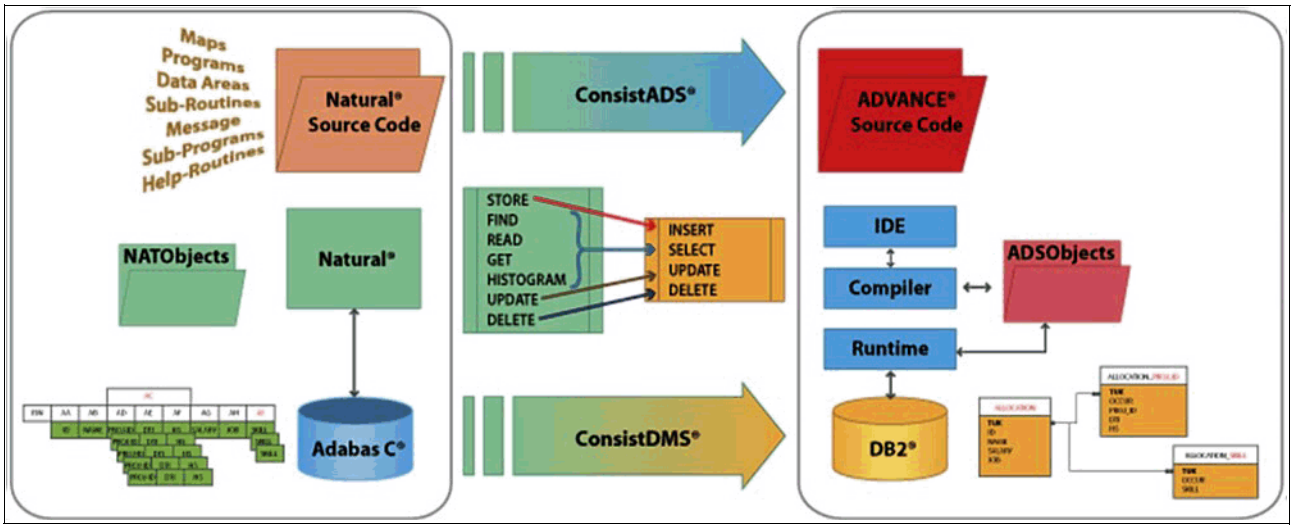

Figure 1 ConsistADS solution architecture

The ConsistADS components are described next.

ConsistADS Mainframe Monitor

The ConsistADS Mainframe Monitor is a z/OS component that provides the centralized operation and coordination of the entire operation involving ConsistADS. It controls the licensing and allows the creation of a network of servers, either in z/OS or in DB2 for Linux, UNIX, and Microsoft Windows machines. Exploring XCF protocol communications, the monitor implements load balancing and disaster recovery allowing the presence of redundant nodes (online servers), either active or in stand by mode.

The mainframe monitor also makes available to **VTAM** the entire set of ConsistADS transactions in effect in every node of the network. This provides integration with CICS.

ConsistADS virtual machine

The ConsistADS virtual machine runs the application programs. The system operates in batch or online modes and includes a mode for batch execution and a mode to run under the ConsistADS Online Server.

The virtual machine performs all of the standard functionality of the languages, such as calculations, file access, device access, database access, and data movement.

ConsistADS virtual machine configuration options are provided through configuration files, which are maintained by the ADSCONF utility.

ConsistADS Online Server

The ConsistADS Online Server is the component that is used to communicate with the web server responsible for the user interface to run online programs. The communication is with a Transmission Control Protocol/Internet Protocol (TCP/IP) set of connections that establish a link between the web server and the server threads.

ConsistADS software operates in a distributed, multiuser, and complex environment and provides a high level of security that is implemented in the solution. Security is provided

through a set of layers that support security constraints to avoid any data breach. The security control is provided through the ConsistADS Online Server interfacing with RACF.

Integration infrastructure

ConsistADS subprograms are integrated through web services. Each subprogram is converted to a Java class, and those Java classes are packaged and accessible through Apache Axis2.

Web Interface Customization Tool

This tool manages maps that are customized by HTML Designer (an editor) and changes the properties of HTML elements. The tool also compares maps to detect the differences between two library versions of the maps and then graphically displays those differences.

ConsistADS IDE

The ConsistADS IDE is a graphical IDE used for software application development and maintenance of source code through a web interface. As Figure 2 shows, the tool provides comprehensive facilities, such as authoring, modifying, compiling, debugging source code software, and deployment of compiled software.

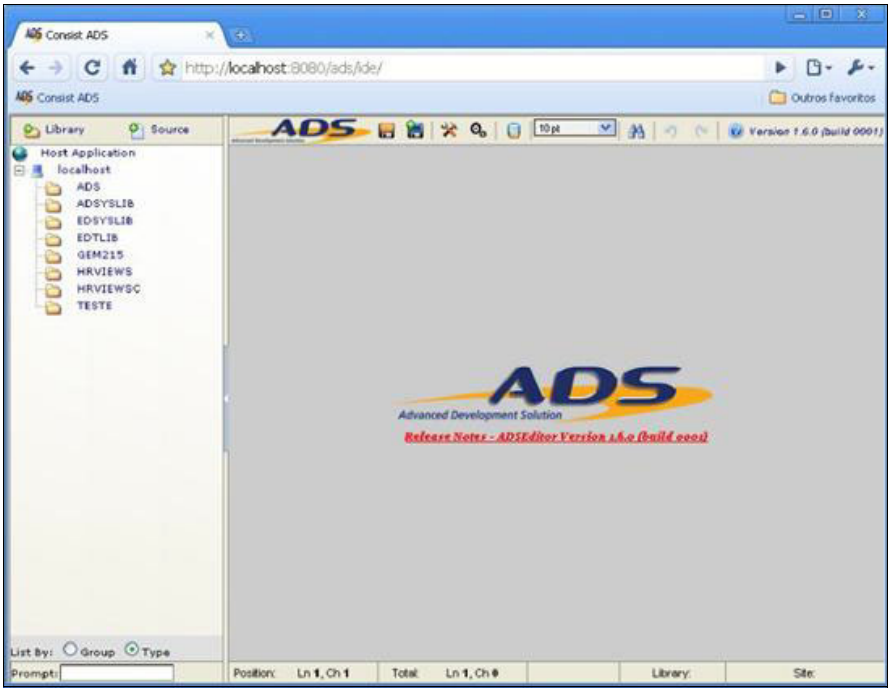

Figure 2 ConsistADS IDE environment view

ConsistADS IDE toolbar

The toolbar (Figure 3) contains the function buttons necessary to develop or maintain source code, including compilation and execution functions.

Figure 3 The ConsistADS toolbar

The ConsistADS IDE toolbar includes the following function buttons:

- The Add Library button () creates a library in the server.
- The New Source button () creates a New Source object for editing.
- The SAVE button () saves the edited object. If it is changed by another user, a confirmation message with the option to rename or change the library is displayed.
- The Save As button () allows a developer to save the new or updated source object with a different name and select a new library, if it is required.
- The Compile button () is used to compile a selected source object, and displaying any messages in the editor footer. If compilation errors occur, a cursor is positioned on the error line and location of the source object.
- The Debug button () is used to start the debugging in a new browser window (tab).
- The Run button () runs the program in a new browser window. The object type determines the run environment. If the object type is a program, the program runs under the terminal context (for example, 3270 on a web UI). If the object type is a web subprogram, the program runs under the web UI context.
- The Deploy button () is used to deploy the binary to a remote server, when the auto-deploy feature is not enabled. By default, when running in z/OS the binary code produced by the compiler is automatically sent to the server for immediate execution, if required by a run or debug command. This feature supports, for instance, generating code and deploying it to more than one runtime environment.
- The DDM Edit button () allows editing DDMs, in its table representation.
- The DDM Mapping Edit button () allows editing the DDM MAPPING file that maps ADABAS fields and files to relational columns and tables.
- The Map Edit button () allows graphic map editing.
- The Find button () allows locating texts inside the objects of a library.
- The Syntax highlighting button (), enables the changing the display attributes of the text to match the Syntax keywords and language elements.
- The Help button () opens the Help panel containing the installed versions, commands allowed in the prompt and the summary of short-cut keys.

ConsistADS Data Migration System

An overview of the ConsistADS data migration process is provided by using the Data Migration System (DMS) tool. This tool complements the information that is provided in the DMS help system and the DMS manual. The tool guides the user through the steps to complete the migration process.

DMS overview

The DMS is a core component of the ADS solution that is used to migrate data from Adabas to a DB2 database. DMS is implemented by using the Common Web Architecture (CWA) environment. DMS contains its own database in which the migration control information is stored and ensures a complete and accurate migration from the original Adabas hierarchical files to the DB2 target database. DMS also provides the specific information that is needed to allow the original application to run and access the new DB2 database.

DMS environment

DMS delivers the components that automatically convert Adabas hierarchical files into a DB2 target database in a single conversion process. The following components are included in DMS:

- DMS data repository

  This repository is the DMS catalog that contains all of the information that is needed for the conversion process. The repository also contains the information about the source and target environments, such as DDMs and File Definition Tables (FDTs). The new data model is stored in the tables of the data repository.

- Project

  The Project concept allows different developers to work with different sets of data in a segregated manner. A Project represents a logical data partition in the DMS Data Repository that corresponds to a set of objects, such as DDMs, FDTs, modeling, tables, and maps, that are related to a specific application or conversion process.

- Logical files

  A logical file defines the mapping of tables and columns between an original Adabas file and the target relational data model. Through logical file abstraction, DMS models Adabas multi-files into several logical views that conform to a new relational database paradigm.

- Modeling

  A developer defines the relational target structure and relationships between the table objects.

  The DMS provides a wizard that defines a standard model that is based on a set of fixed rules. For example, each periodic group has a correspondent child table that includes all of the periodic fields. The super-descriptors are defined as extra fields in the main table.

  In addition, DMS offers a modeling process that is customized. A developer defines a set of conversion rules to allow different target specifications for the same original field or group of fields. For example, one developer might want to keep the contents of a multi-valued field in a single column in the relational table. Instead, the developer might choose to split the first five occurrences of a particular multi-valued field into five different fields (table columns).

- Catalog tables

  Catalog tables contain the complete target relational database structures and all of the information about the target DB2 environment, such as tables and columns, keys, referential integrity, and index specification.

- Mapping

  This component generates the map file DDMMAPPING that is used by the ADS compiler to translate Adabas commands to SQL statements.

- Models and templates

  Several parts of the conversion process are done through script execution by using JCL. DMS generates these JCL scripts that are run during the migration process.

- Batch subsystem

  The DMS system uses the CWA batch subsystem environment to conduct several conversion tasks, such as, importing and generating. Those batch procedures are started automatically and are analyzed by using the CWA batch processing options.

- Source environment

  This environment contains the Adabas/Natural environment in which the data is unloaded and converted. The environment also includes the FDTs and DDM from the source environment that used in the modeling process. The DMS system generates all of the JCL scripts for the unload process and the conversion programs that arrange the data according to the new database relational model.

- Target environment

  The new DB2 relational database data is loaded into the target environment. All DDL that is required to create the target database, table objects, and JCL scripts that are used in the load process are generated by the DMS system.

- Server environment

  The ConsistDMS system works under the CWA environment of Consist so that the user runs client/server applications through the web.

- Local environment

  The web browser runs under the local Linux for System z or z/OS UNIX System Services (USS) environment.

- Conversion programs

  These programs are a set of programs that are automatically generated to convert the data from the source model (Adabas) to the target model (relational) according to the data modeling specified for each file. The conversion programs read the unloaded and decompressed data from the Adabas environment to generate the sequential files that are used as input for the corresponding DB2 load utility. All of the JCL that is required to run these batch programs are automatically generated by DMS.

- Validation programs

  The validation programs are a set of programs automatically generated for data conversion validation. These programs read both Adabas source data files and the new DB2 tables to generate sequential files that are used to compare source and target data files.

- ADS application environment

  This ADS Library structure includes the system libraries necessary to manage the ADS environment.

How ConsistADS maps Adabas data files to DB2

Adabas has its own database design, such as, data structures or objects. When you convert a database from Adabas to DB2 for z/OS, the database objects are converted into optimized DB2 table/index objects. The Natural language programs access the Adabas database with different command-calls that need to be translated to SQL calls. These conversions are described in this section.

Adabas concepts: The specific concepts of Adabas are not covered in this book. The focus of the book is on the overall conversion process.

Data design

In this section, we examine data constructs mapping from Adabas to DB2.

Adabas file layouts

The Adabas files are converted to DB2 tables according to the FDT definition.

For the DDMs, the conversion tool creates the VIEWs on DB2.

When Adabas data fields are translated as DB2 columns, the columns must adhere to the following rules:

- Column names are unique.
- Redefines are not used.
- Columns are defined as NULL, NOT NULL, or a default value.
- Fixed data and user VARCHAR fields must be reviewed to avoid unnecessary space.

Group-level elements (MU and PE occurrences)

Prerelational databases often use groups of repeating data in the same field to save disk space for data and index pointers. Figure 4 shows these groups of repeating data inside an Adabas file.

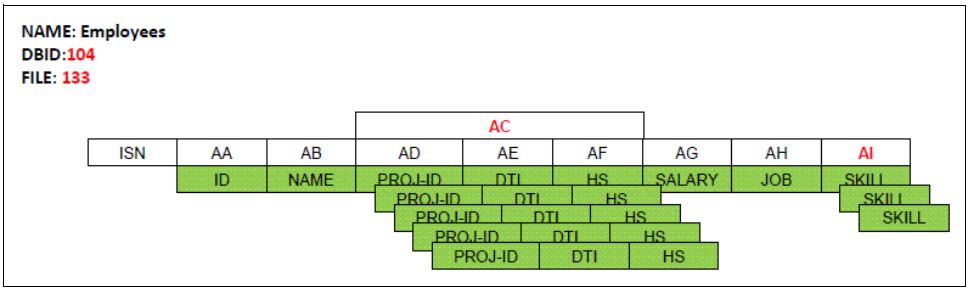

Figure 4 Adabas file design

Several approaches are used to resolve these repeating groups by using a relational model. Figure 5 shows the relational model that was normalized for these groups of repeating data, which results in new DB2 tables. The tables require foreign keys to establish all of the relationships that are declared in DB2 by using the referential integrity rules.

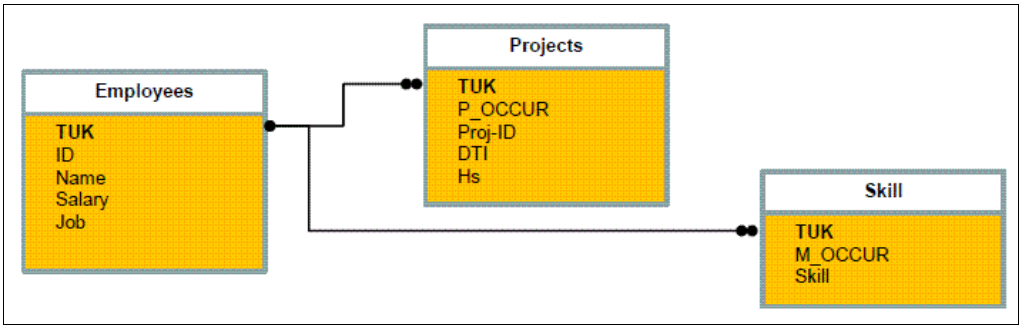

Figure 5 Normalized model

The primary and foreign keys are defined in DB2 to maintain those relationships that are created in Adabas. If the keys exist, the keys are left in the code for the conversion.

Referential integrity: For more information about referential integrity in DB2 for z/OS, see *Data Integrity with DB2 for z/OS*, SG24-7111.

Other options are available to use according to performance needs and data modeling denormalization rules.

Figure 6 shows a denormalized data model.

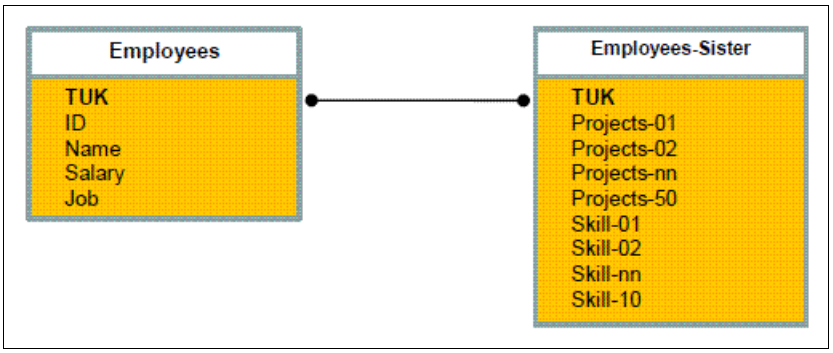

Figure 6 The denormalized model

Another option is to use a fully denormalized data model. However, this option causes performance issues, as shown in Figure 7.

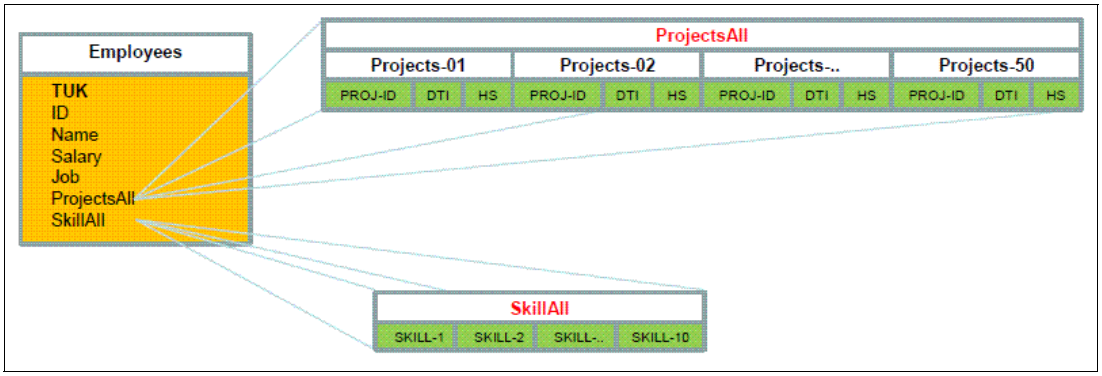

Figure 7 Fully denormalized model

Super-descriptors

Super-descriptors are the equivalent of indexes in DB2 for z/OS. A super-descriptor is a new index that is created and used in programs as search criteria.

Application design

Adabas supports several procedural languages, such as COBOL, by using ADASQL and direct calls. Natural is the most commonly used language, so we focus more on this language in this section.

Example 2 shows the conversion for GET ISN, in which the accesses are made physically to the record location.

Example 2 GET ISN conversion

```
GET *ISN or READ ISN = value

SELECT FROM ... WHERE ROWID = value
```

Example 3 shows the use of the HISTOGRAM command.

Example 3 HISTOGRAM conversion

```
HISTOGRAM ...

SELECT DESCRIPTOR COUNT(DESCRIPTOR)
WHERE DESCRIPTOR => value
GROUP BY DESCRIPTOR
```

The ORDER BY clause might be added also.

Table 2 shows how Natural commands in the application are matched to SQL statements in DB2.

Table 2 Natural DML compared to DB2 SQL

Natural DML	SQL
FIND	SELECT
FIND FIRST	SELECT, FETCH FIRST 1 ROW ONLY
FIND UNIQUE	SELECT
FIND SORTED	SELECT... ORDER BY
READ … LOGICAL	SELECT … ORDER BY
GET, GET SAME	SELECT
READ … PHYSICAL	SELECT
HISTOGRAM	SELECT … GROUP BY
STORE	INSERT
UPDATE	UPDATE … OF CURRENT CURSOR
DELETE	DELETE … OF CURRENT CURSOR
END TRANSACTION	COMMIT
BACKOUT TRANSACTION	ROLLBACK

Many SELECT calls require a CURSOR on DB2 for z/O. Cursors must be disabled when only one row is retrieved.

The ORDER BY clause requires sort use, unless indexes are used in the columns that are referenced by the ORDER BY clause.

An SQL tuning team must check the DB2 access path, index use, DB2 resources, and so on.

DB2 for z/OS programming: For more information about DB2 for z/OS programming, see *DB2 10 for z/OS Application Programming and SQL Guide,* SC19-2969.

ConsistADS conversion process

In this section, a case study example of a ConsistADS conversion is reviewed.

Customer's challenge

The customer wanted to upgrade their System z servers, modernize their IT environment, and reduce costs.

The customer required the conversion of the Natural applications that are listed in Table 3 on page 17 and the Adabas files that are listed in Table 4 on page 17.

Table 3 Natural applications

Application name	Program	Lines of code
APP1	2574	637885
APP2	520	46281
APP3	489	127884
APP4	75	13100
APP5	1105	266242
APP6	1405	452458
APP7	411	80825
APP8	167	40860
APP9	3682	767663
APP10	357	53771
APP11	225	18091
APP12	304	54861
APP13	812	136671
APP14	202	10474
Total	12328	2707066

Table 4 Adabas files

DB ID	Files	Size GB	Data GB	Millions of records
3	176	330	210	1397.0
30	5	123	75	869.0
10	52	6.3	4.1	13.7
15	72	16.3	7.2	30.8
Total	305	475.6	296.3	2310.5

Solution architecture

The ConsistADS under z/OS allows several implementation options, always involving the ConsistADS monitor, which controls the licensing and the cooperation of one or more online servers. The entire set of possibilities is illustrated on Figure 8 on page 18.

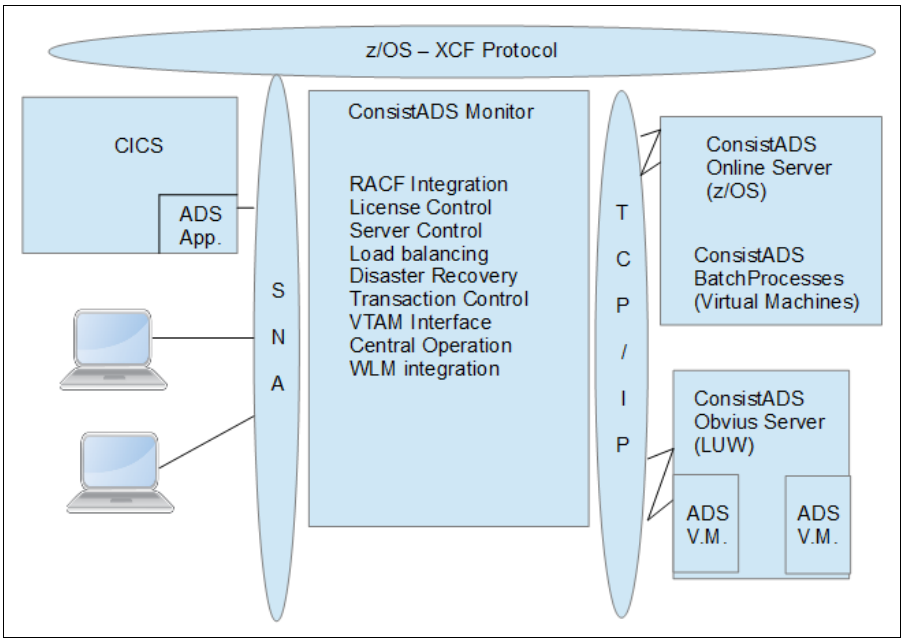

Figure 8 Consist Online Server

Figure 8 shows the general overview of the multi-server solution using ConsistADS. The monitor component is responsible for the License Control and serves as a central repository for licenses over the entire installation. This component also includes the Transaction Server which can be used to interface with VTAM terminals and other z/OS components such as CICS.

The monitor accesses the ConsistADS application servers located in z/OS, z/Linux or other LUW platforms. This access is possible because the ConsistADS architecture uses TCP/IP as the standard for interfacing with the end-user as well as any other process that communicates with it. Due to this characteristic, ConsistADS can communicate with Natural RPC, and standard Web-services. AXIS2 objects can directly access the objects that run inside those VM Containers.

Further, the monitor can be installed in any LPAR not necessarily related to the Application Server LPARs. The XCF protocol implements the support for sysplex, allowing multiple distinct and independent servers to be active at the same time.

With details of one of the implemented logical partitions (LPAR) in z/OS, a typical production environment is shown in Figure 9.

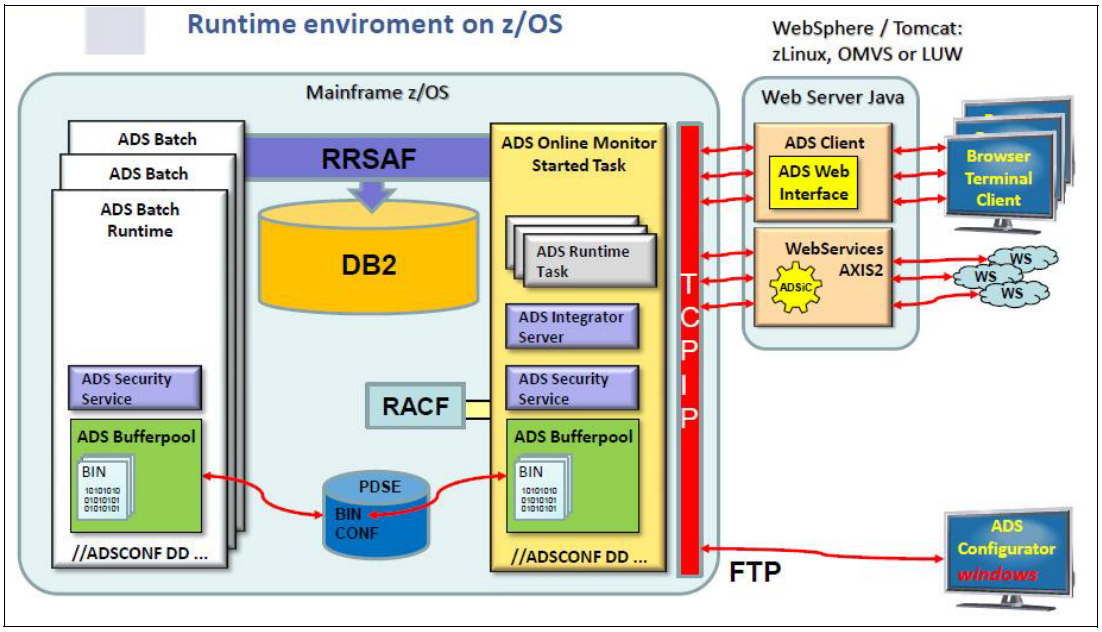

Figure 9 Runtime environment

Figure 9 illustrates the ADS functionality of each application server environment in a z/OS LPAR. As mentioned before, the ConsistADS z/OS Monitor does not need to be located in the same LPAR as the ConsistADS application servers. The figure just exemplifies this possibility to illustrate the support for 3270 user interface under control of it (what is “it”?).

The virtual machines can run independently in the case of batches, and under the control of the application server which implements the TCP/IP layer responsible for the all types of communication.

The ADS Web Protocol is used to distribute the online and integration features over the entire network. So the figure shows not only users for online interface, but also, developers that are testing their applications and the Web-services that are consuming the functions provided by the application in z/OS.

The converted Natural language, Advance, runs over the runtime environment in batch mode and online. The interface uses the web server to show web windows instead of showing the Natural maps.

Project phases

The entire project took approximately six months to complete. Table 5 shows the amount of time spent on each phase of the project.

Table 5 Project timing

Phase	Amount of time
Performing environment analysis	One month
Executing data conversion	One month
User unit testing	Two months
User quality assurance testing	One month

Phase	Amount of time
Define JCL streams and batch jobs integration	Two months (conducted in parallel with the two previous phases)
Final user testing and training	One month
Turn over into production	Two days (weekend)

The customer decided to implement each application as described in the methods for the conversions.

Unique challenges: Each customer environment could pose unique challenges that might affect timing for the project.

The steps of the project included the following details:

Project initiating process

- Develop the project charter
- Develop stakeholder management strategy

Project planning process

Develop project management plan.

Project running process

- Collection and analysis of the following information and requirements at the client site:
  - Answer questionnaire
  - Analysis of online applications
  - Database analysis
  - Premises compliance by client
  - Analysis of batch applications
  - Preparation of maps (linking business services with applications)
  - Preparation of diagrams that describe the network configuration
  - Back up generation of initial application
  - Back up generation of initial DB
  - Back up generation: JCL, PROC, and PARAM
  - Environment of migration at Consist:
    - First cycle of migration - automatic
    - Research and JCL adaptation
    - General tests of first cycle
    - Second cycle of migration - refinement
    - Migrated object packing
  - Quality assurance environment at the customer site
    - Preparation of the work infrastructure z/OS (LPAR, DB2, RACF, and ADS)
    - Preparation of the work infrastructure Linux for System z (Apache Tomcat, Java, and ADS)

- Test plan development
- ADS installation
- ConsistADS mainframe installation
- ConsistADS Linux installation
- Installation of application converted
- DB2 for z/OS installation
- SMP/E work
- DB2 subsystem installation
- DB2 Administration Tool
- Activate DB2 Administration Tool
- Data loading
- Web services integration
- User access settings
- ADS training to client (systems analysts)
- Process tests (systems analysts) and adjustments
- ADS training to client (users)
- Process tests (systems analysts) and adjustments
- Performance tests and adjustments
- Stress tests (online and batch applications) and adjustments
- User training
- Production environment at client, which includes ADS installation, DB2 for z/OS installation, converted application installation, and data load migration

– Monitoring and adjustments in production
  - Follow-up and final adjustments

Project controlling process

- Project status report
- Monitor and control the project

Project closing process

- Run project acceptance
- Develop and document lessons learned

Data conversion

The ConsistADS solution uses the DMS to convert Adabas files to DB2 tables. DMS maps the Adabas calls to SQL commands to access DB2 for z/OS.

DMS contains its own database in which the control information is stored. The contents of DMS ensures a complete and accurate migration from the original Adabas to the target DB2, and allow the original applications to access DB2.

Figure 10 shows the flow of the steps to conduct the data conversion.

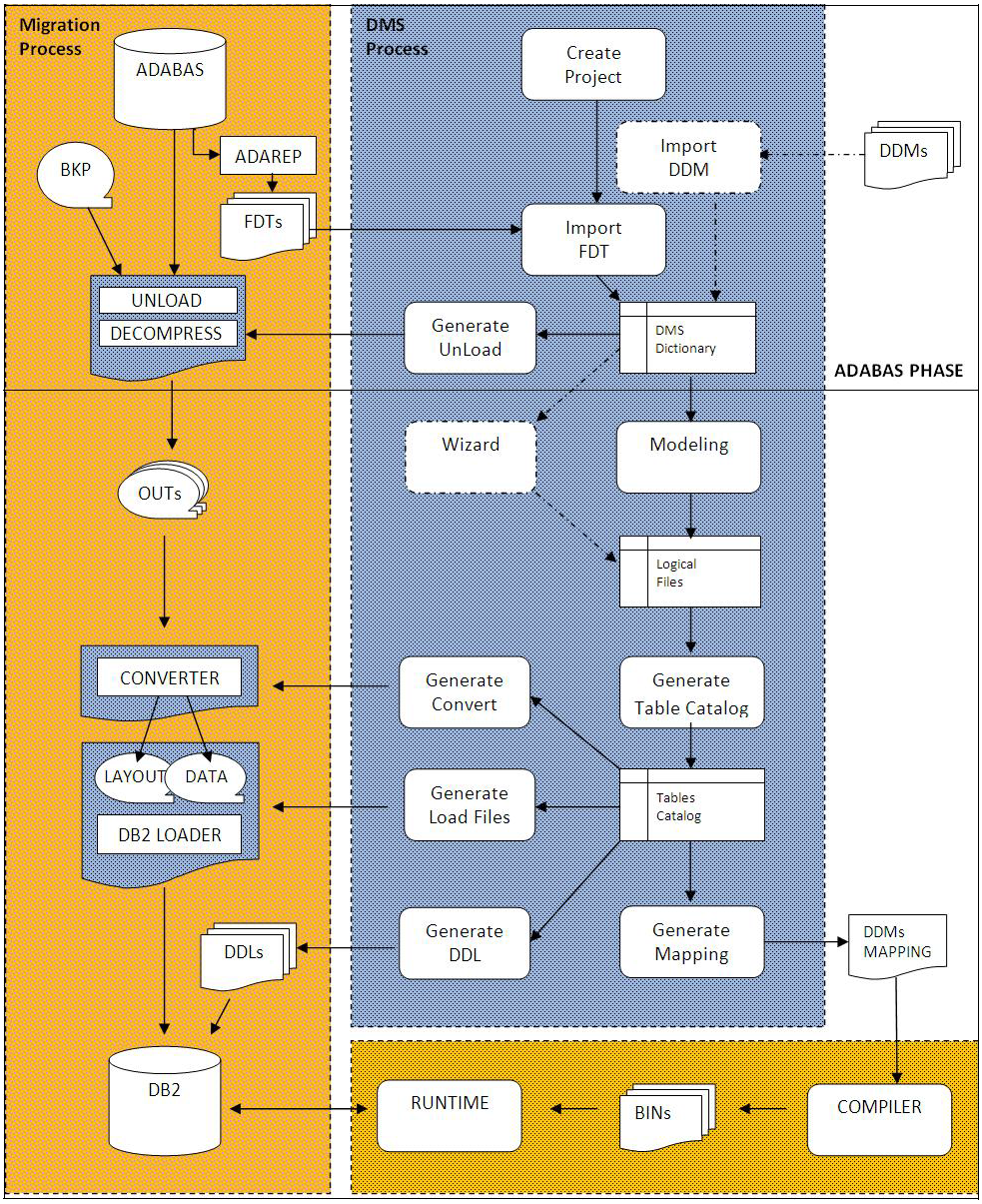

Figure 10 DMS tasks

This section describes the steps shown in Figure 10:

1. Generate ADAREP from the customer's environment and gather all Adabas information, such as the FDT and DDMs.
2. Create a project work flow, including iterations for the source and target databases.
3. Import FDTs.

   The FDTs used in the import process must be generated in the source environment by using the Adabas Report utility. The Report output must be made available either in the local or server environment for use by the import utility.

   The imported files are listed after the import process is complete, as shown in Figure 11.

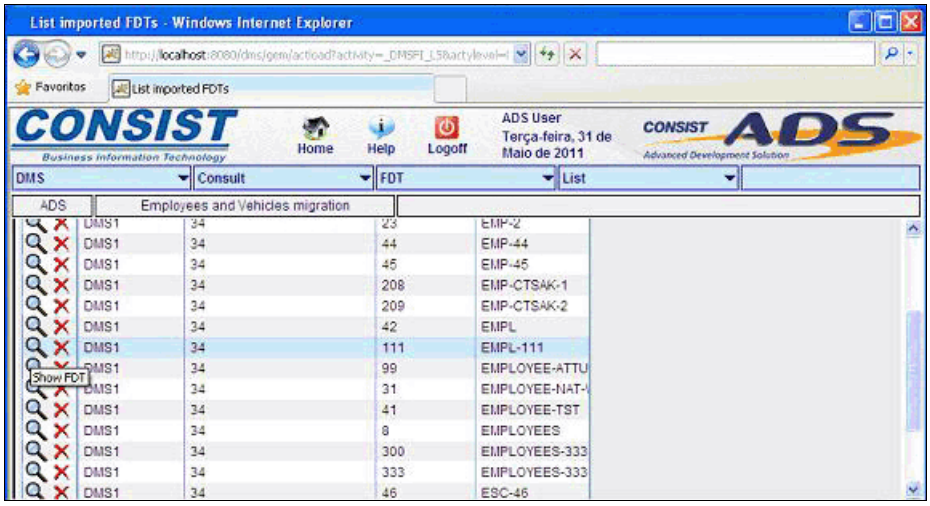

Figure 11 List of imported files

The files are opened and inspected, as shown in Figure 12.

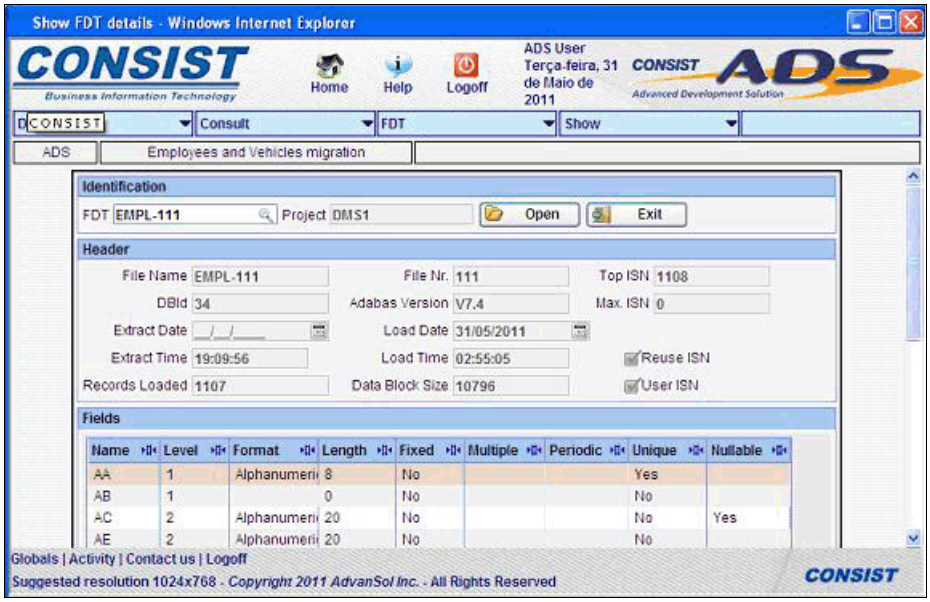

Figure 12 Details of the file attributes

Importing DDMs

DDMs are the equivalent of the table views in DB2 for z/OS.

The DDMs must be converted to text format through the DMS Source File Migration function before the DDMs are used by the IMPORT utility. The DDMs must be extracted from the Natural production environment by using the SYSOBJH utility (Transfer Mode). The DDMs are then transferred to the source objects in the development environment by using FTP.

Figure 13 shows the DDM listed in DMS.

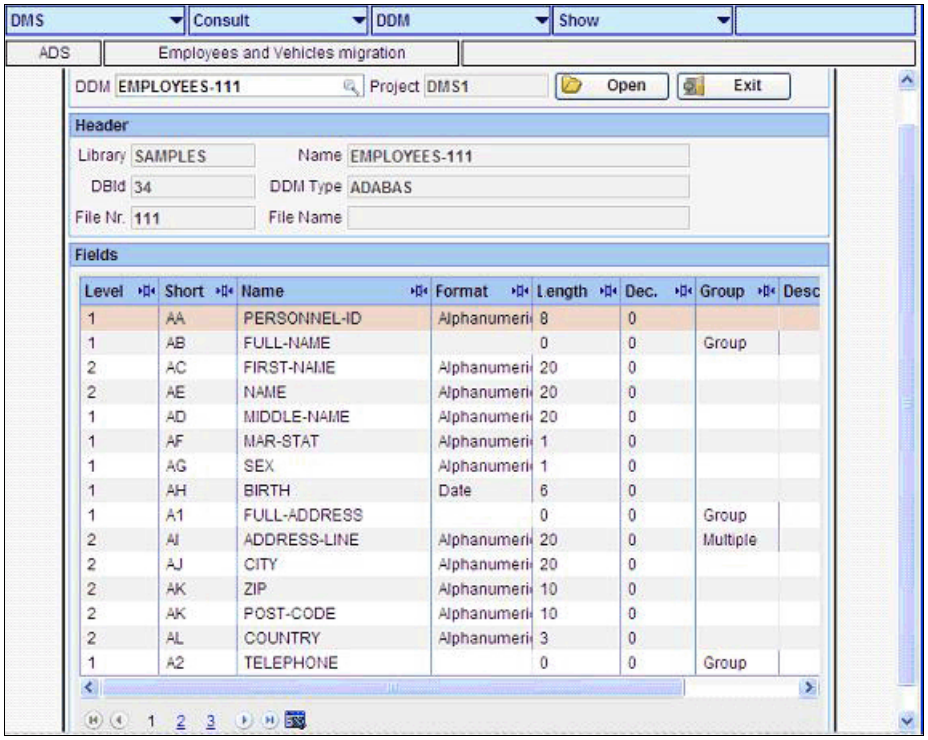

Figure 13 DDM imported

Modeling

After the FDTs and DDMs are imported into the DMS system, the modeling process is started for each imported FDT.

Each modeling process that is run generates a logical file that represents the correspondence between the original Adabas file and the new relational data mode. Adabas allows for a single FDT to support several DDMs, and a single field in the FDT is reused within several DDMs that serve as the logical record criteria to extract data into DB2.

A logical file defines the relationship between the original Adabas file and the target relational data model in terms of tables and columns. Through the logical file abstraction, we model Adabas multi-files into several logical views in a relational database paradigm.

Figure 14 shows the correlation of the DDMs to a data model.

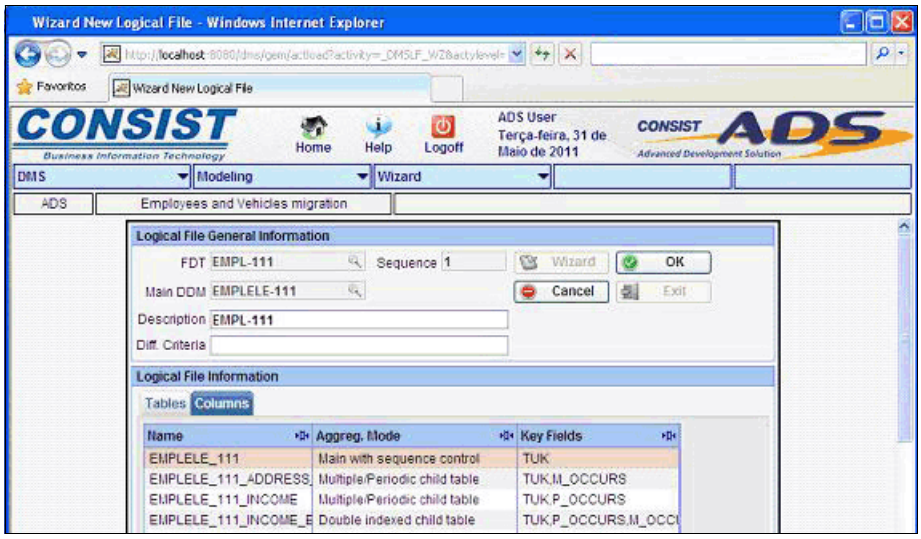

Figure 14 DDM and corresponding data model

Figure 15 shows the relationship of DDMs fields to each column in the target table.

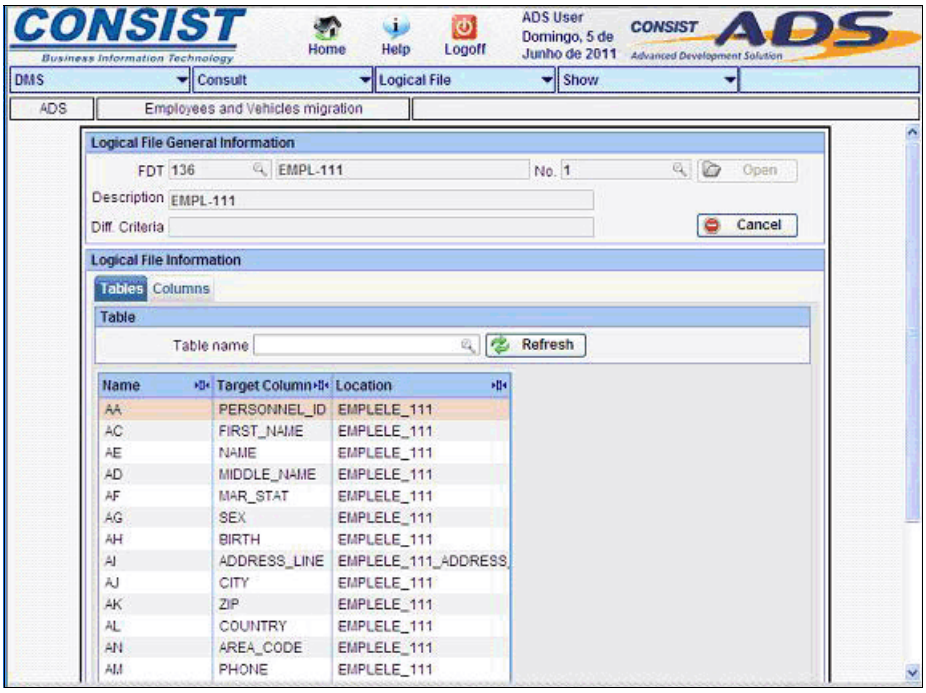

Figure 15 Logical file

Generating the table catalog

Generating the table catalog is an important function because the metadata tables of the Target Environment in the DMS Data Repository are generated. These tables contain all of the information about the data modeling, which represents the new relational data organization. The DMS Data Repository is extensively used by the data conversion process, and during DDL generation and load process mapping.

To start the process, you select an FDT (the Adabas model) and the corresponding logical file (the new relational model), as shown in Figure 16.

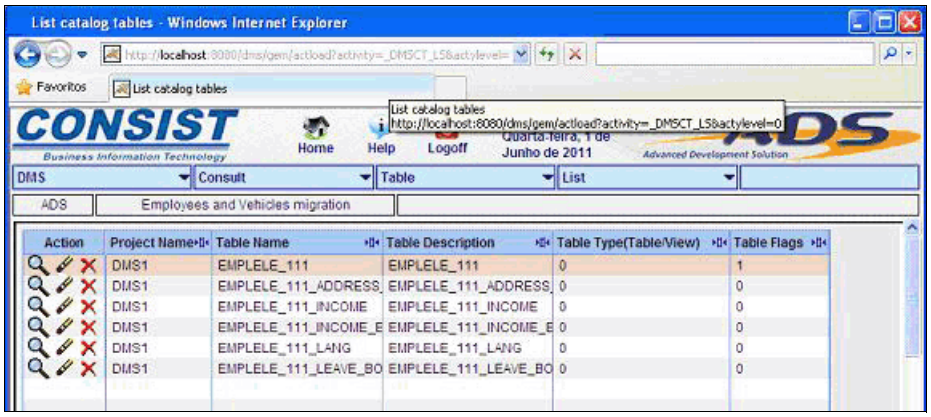

Figure 16 Generating DDL

The user generates the DB2 DDL for each logical file, as shown in Figure 17.

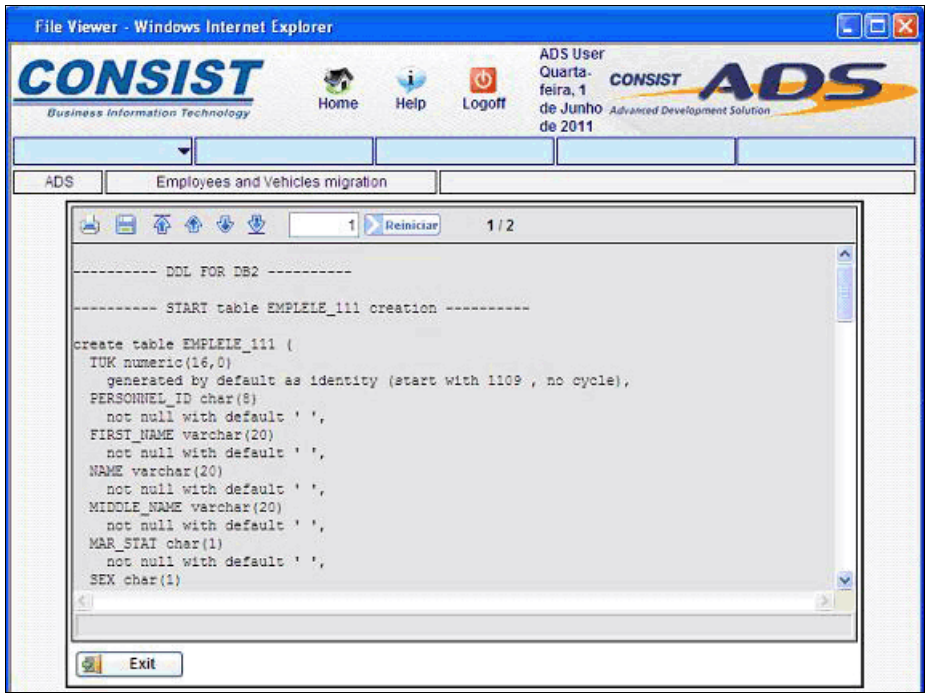

Figure 17 DB2 for z/OS DDL

Mapping

The Mapping function generates the `DDMMAPPING` file, which is used by the ADS compiler to replace all of the Adabas call statements with SQL. This replacement is an important component of the application conversion process.

Figure 18 shows the interface that is used to conduct this process.

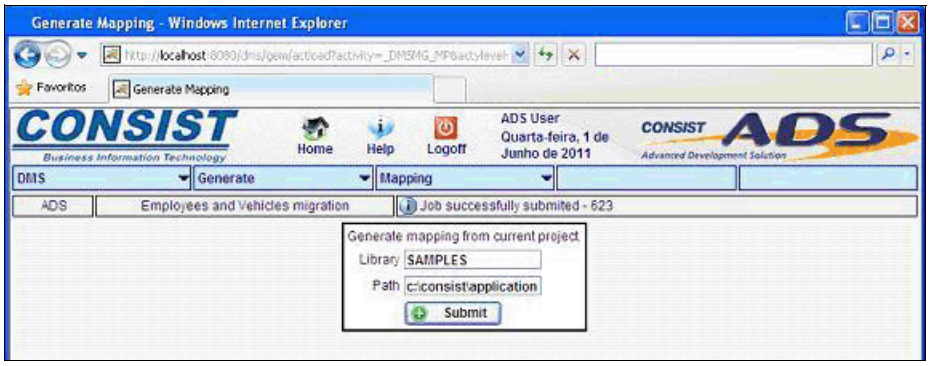

Figure 18 Generating mapping

Generating the conversion and validation programs

This function generates the conversion and validation programs. The programs must be cataloged by using the ADS compiler. The corresponding objects are transferred to the target environment for subsequent execution, as illustrated in Figure 19.

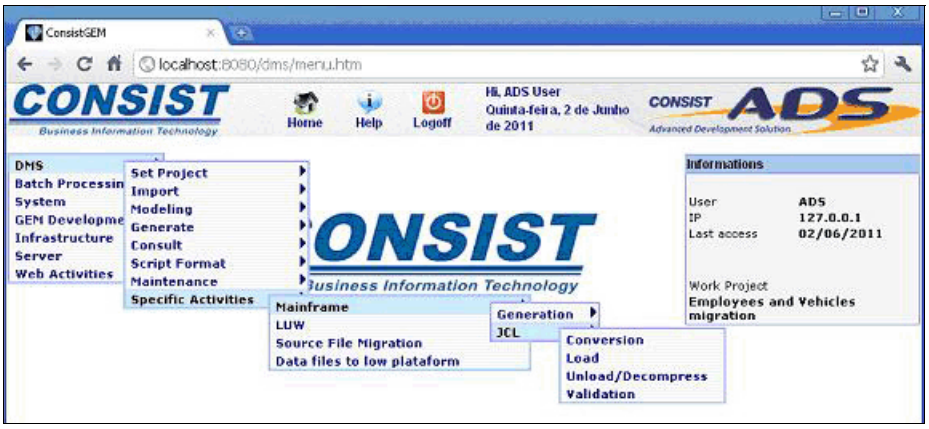

Figure 19 Generating JCL to conversion

This function generates the JCL necessary for the execution of the Conversion and Validation programs as part of the migration process on the source or the target environment.

The following groups of JCL statements must be generated:

Conversion Generates the programs to create the unload data

Unload/Decompress Unloads the Adabas by using the new data modeling

Load Inputs the data into DB2 for z/OS

Validation Validates the process

After the generation process is complete, the JCL statements must be transferred to the source and target environments for execution.

Figure 20 shows an example of selecting a file to generate the unload process. The customer provides the number of occurrences and the arguments for space allocation on the appropriate fields.

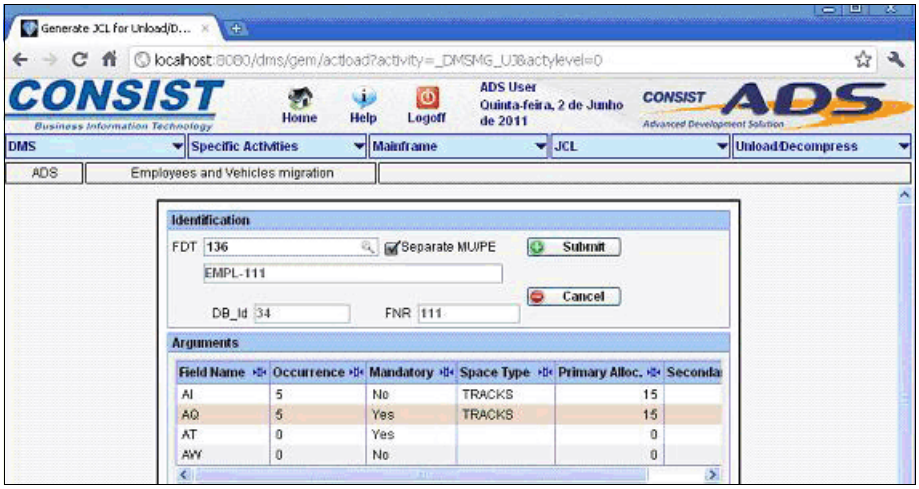

Figure 20 Creating unload process

The JCL that is run in z/OS environment is shown in Figure 21.

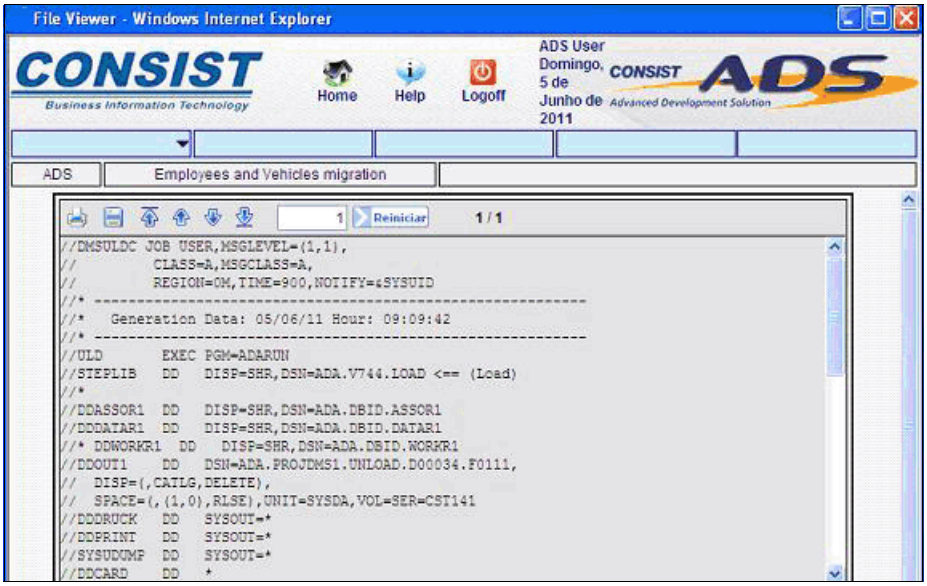

Figure 21 Generated JCL

After these steps are completed, it is possible to start the conversion of the Natural language as described in the next section.

Converting the application

To start the code conversion from Natural to Advance, you must send the SYSOBJH to generate all of the Natural source codes. ConsistADS reads the source code and generates the binary object in ADS with the SQL statements.

Example 4 shows the source code of a Natural program before the conversion.

Example 4 Natural source program

```
define data local
  1 v view of SPE-NOTIFY-CAMREG
    2 CAN-NSS-OFI
  1 counter (i4)
  1 cont2   (i4)
  1 read1   (i4)
end-define
write  'PGM - BEGIN'
reset counter cont2 read1
read v by isn
  add 1 to read1
  if CAN-NSS-OFI >= 0 then
    Update
     add 1 to  counter
     add 1 to cont2
    if counter = 1000 then
     commit
    write 'commit record:'  cont2 ' read nreg:' read1
    reset counter
  end-if
 end-if
end-read
write 'FINAL NREG:' cont2 'read nreg'read1
end
```

ConsistADS compiler generates dynamic or static SQL by creating assembler modules.

Example 5 shows the generation of dynamic programs.

Example 5 Program converted to dynamic Advance language

```
1 v view of SPE-NOTIFY-CAMREG
    2 CAN-NSS-OFI
  1 counter (i4)
  1 cont2   (i4)
  1 read1   (i4)
end-define
write  'PGM - BEGIN'
reset counter cont2 read1
ADS_LB1.
SELECT CAN_NSS_OFI,SPE_EMP_NRP,SPE_TPO_REG,SPE_EMP_NRP,
  INTO V.CAN-NSS-OFI,
       SPE_SUP_NSS_NRP.SPE_SUP_NSS_NRP_SPE_EMP_NRP,
       SPE_SUP__TPO_NSS_NRP.SPE_SUP__TPO_NSS_NRP_SPE_TPO_REG,
       SPE_SUP_TPO_NSS_NRP.SPE_SUP_TPO_NSS_NRP_SPE_EMP_NRP
  FROM SPE_NOTIFI_CAMREG ORDER BY SPE_NOTIFY_CAMREG.TUK
       FOR UPDATE OF CAN_NSS_OFI,SPE_SUP_NSS_NRP,SPE_SUP_TPO_NSS_NRP
   ADD 1 TO READ1
   IF V.CAN-NSS-OFI >= 0
    UPDATE SPE_NOTIFI_CAMPREG SET CAN_NSS_OFI = V.cAN-NSS-OFI,
           SPE_SUP_NSS_NRP=SPE_SUPNSS_NRP.SPE_SUP_NSS_NRP_B,
           SPE_SUP_TPO_NSS_NRP.SPE_SUP_TPO_NSS_NRP_SPE_EMP_NRP_B
```

```
    WHERE CURRENT OF CURSOR(ADS_LB1)
    ADD 1 TO CONTADOR
    ADD 1 TO CONT2
  IF COUNTER = 1000
     COMMIT
   WRITE 'commit nreg:' CONT2 'read nreg:' READ1
   RESET COUNTER
  END-IF
 END-IF
END-SELECT
WRITE 'FINAL :'CONT2 'read nreg:'READ1
END
```

Example 6 shows the generation of static assembler modules.

Example 6 Program converted to Static Assembler

```
*===========================================
* Object   : PGM
* TimeStamp: NRUNSTPL13514194
*===========================================
ADSWK01  DSECT
*
*===========================================
*     Parameters
*===========================================
P0000001 DS    P'99999999999'
P0000002 DS    CL17
P0000003 DS    CL18
*
*===========================================
*     Variables
*===========================================
V0000001 DS    P'99999999999'
V0000002 DS    P'99999999999'
V0000003 DS    CL11
V0000004 DS    P'9'
V0000005 DS    CL11
*
*===========================================
*     SQL cursors
*===========================================
*
         EXEC SQL DECLARE CADS0001 SENSITIVE STATIC SCROLL CURSOR WITH X
               HOLD FOR SELECT SPE_NOTIFI_CAMREG.TUK,CAN_NSS_OFI,      X
               SPE_EMP_NRP,SPE_TPO_REG,SPE_EMP_NRP FROM                X
               SPE_NOTIFI_CAMREG ORDER BY SPE_NOTIFI_CAMREG.TUK FOR    X
               UPDATE OF CAN_NSS_OFI,SPE_SUP_NSS_NRP,                  X
               SPE_SUP_TPO_NSS_NRP
LBOP0001 DS    0H
         EXEC SQL OPEN CADS0001
LBFT0001 DS    0H
         EXEC SQL FETCH CADS0001 INTO :V0000001,:V0000002,:V0000003,   X
               :V0000004,:V0000005
LBCL0001 DS    0H
         EXEC SQL CLOSE CADS0001
```

```
*
         EXEC SQL UPDATE SPE_NOTIFI_CAMREG SET CAN_NSS_OFI=:PO000001,  X
               SPE_SUP_NSS_NRP=:PO000002, SPE_SUP_TPO_NSS_NRP=         X
               :PO000003 WHERE CURRENT OF CADS0001
*
*=========================================
*     Finalize
*=========================================
         EXEC SQL INCLUDE SQLCA
*
         END
```

As shown in Example 4 on page 29, Example 5 on page 29, and Example 6 on page 30, you use the programs to access DB2 for z/OS dynamically or in static mode.

The DDMMAPPING function, as described in “Mapping” on page 26, is a critical step to accurately generate the application code. Example 7 shows the DDMMAPPING for the code.

Example 7 Mapping for the source programs

```
[FILE=TABLES]
SPE-NOTIFI-CAMREG = SPE_NOTIFI_CAMREG
[FIELDS]
CAN-NSS-OFI        = SPE_NOTIFI_CAMREG.CAN_NSS_OFI
SPE-EMP-NRP        = SPE_NOTIFI_CAMREG.SPE_EMP_NRP
SPE-TPO-REG        = SPE_NOTIFI_CAMREG.SPE_TPO_REG
SPE-SUP-NSS-NRP    = SPE_NOTIFI_CAMREG.SPE_SUP_NSS_NRP
SPE-SUP-TPO-NSS-NRP= SPE_NOTIFI_CAMREG.SPE_SUP_TPO_NSS_NRP
```

Once the application is compiled, both batch and online processes are run in the ADS runtime environment, as shown in Example 5 on page 29 and Example 6 on page 30.

ConsistADS client results

From the business perspective, the customer sees the following results after the conversion project is completed:

- Maintenance costs are reduced.
- Multi-tier architecture and SOA are enabled to improve agility for IT to deliver business requirements.

Table 6 summarizes the results of the conversion.

Table 6 Results

Before conversion	After conversion
Mainframe model 2096/S03	Mainframe model 2817/504
3 CPUs	3 CPUs
12 GB memory	5 GB memory
2 terabytes on disks	5 terabytes on disks
20 MSUs	20 MSUs
z/OS 1.6	z/OS 1.10
Complete 6.3.1	ADS monitor
Natural 3.1.6	ADS runtime
Adabas 7.1.3	DB2 for z/OS version 9
EntireBroker 7.2	ADS integrator
Natural connection	ADS web interface
Adabas Security 7.1.3	ADS access control / DB2 privileges
Natural Security 3.1.6	ADS security
305 Adabas files	459 DB2 tables
2310 millions of records	4859 millions of records
14 applications with 2.7M lines of code (13000 online programs and 600 batch programs)	Same
350 Online Users (with an average of 80-125 concurrent users)	Same
1 Main Office site and 120 remote office sites	Same

Author

This paper was produced by a team of specialists from around the world working with the IBM Redbooks team.

Paolo Bruni is an IBM Redbooks Project Leader who is based at the Silicon Valley Lab in San Jose, California, in the US. Since 1998, Paolo has authored IBM Redbooks publications about IMS, DB2 for z/OS, and related tools and has conducted workshops worldwide. During his many years with IBM in development and in the field, Paolo's work has been related mostly to database systems.

Clauder Balzano is an experienced IT specialist that has participated in and led numerous projects involving Software AG's Adabas & Natural and EntireX on IBM z/OS and other IBM platforms. His work has taken him to many countries around the world giving him a broad understanding of the needs of the IBM z/OS client base and their migration needs. Most recently Clauder has worked on three continents migrating SAG clients to IBM DB2 z/OS. Clauder is also experienced with AIX, HP-UX, Oracle Solaris, Microsoft Windows and Linux technologies.

Now you can become a published author, too

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time. Join an IBM Redbooks residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks
- Follow us on Twitter:

  http://twitter.com/ibmredbooks
- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806
- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm
- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®	OMEGAMON®	System z®
CICS®	RACF®	Tivoli®
DB2®	Redbooks®	VTAM®
IBM®	Redbooks (logo)  ®	z/OS®

The following terms are trademarks of other companies:

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

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