

COBOL/VSAM TO JAVA

VSAM files and SQL Database is often used as data-stores in COBOL applications which do not utilize transaction-processors. This document will outline how each one is migrated within SoftwareMining's COBOL to Java / C# approach.

INDEXED & KSDS FILES TO SQL

Whilst many COBOL applications have moved away from use of VSAM files to relational databases ([COBOL with Embedded SQL](#)), or use transaction handle libraries (e.g. [CICS](#)), there are still many COBOL applications which still utilize VSAM KSDS file for their data. The best approach to migration of VSAM structures would map every field to a different column. Complication could arise when dealing with complicated REDEFINITIONS or multi record File-DEFINITIONS - but a good design will often can simplify and cater for such issues very quickly.

Other issues which need to be addressed is when data is scattered across multiple files (e.g. one data-files per year), GDG support and other issues.

Performance of java layer for accessing the database is also a primary concern: Whilst SQL database has better performance for searching across large data-sets, the retrieval of data may be slower as data is often have to be transmitted across a network (database server rarely lies on the same machine - all communication has to go via TCP). As such a good design for translated Java/C# code, the database, as well as the data-access libraries is essential.

SEQUENTIAL FILES

Sequential files are often used as intermediary means of saving data. They are sometime used in communication with other programs, for sorting or for report generation. SoftwareMining translations by default do not migrate these to SQL tables, but continue maintaining these on file-system - although it is very easy to also write them to SQL databases.

The Sequential files generated by SoftwareMining benefit from following features:

- Binary Compatible with files generated on COBOL System. This allows them can continue to be used for communication with remainder of the system (the as yet untranslated modules)
- Provide easy migration path to new standards - such MS Excel

RELATIVE FILES

Relative files may be maintained as files within file system, or relational database tables.

MULTIPLE FILE-DEFINITION FOR SAME DATA-FILE.

In an application, often different programs define a different structure (FILE-DEFINITION) for same datafile.

Whilst such approach causes no problems for when dealing with (data) Files - is not a viable approach for defining columns in database. The manual consolidation or merge of such structures can be lengthy and error prone. Automatic identification and conflict/resolution of these definitions may be one of the most important aspect of a successful migration projects.

PERFORMANCE , PERFORMANCE , PERFORMANCE

Batch programs often have to complete their processing within a set timeframe. The performance issue has to be one of the major influences on design of the translated java or C# code.

The following can have very large impact on performance of migrated code:

- Number fields: Where possible and **safely** , the translator should convert COMPUTATIONAL formats to native java data-types
- Precision: Java BigDecimal should be used only when absolutely necessary
- Remove un-used structures - cut down on load time
- Parallelization - is achieved much easier on legible and maintainable code (translated Java/C# code).
- REDEFINES - can the translator identify and remove any of them

MIGRATION OF COBOL WITH EMBEDDED SQL TO JAVA OR C#

Many COBOL applications have moved away from use of VSAM files to relational databases SQL.

This step would have significantly improved the data-maintenance of the application as well as easing integration with other non-COBOL applications.

COBOL ESQL applications are hence the prime candidates for migration to Java or C#. This path can bring the general application maintenance inline with the already improved data maintenance.

It is worth noting that ESQL applications are also prime candidates for [gradual and migration projects](#).

COBOL WITH EMBEDDED SQL MODERNIZATION

Some COBOL program use DB2 instead of VSAM files. Access to DB2 is provided via DB2 SQL queries. Migration of EMBEDDED SQL Programs involves the translation of DB2 SQL to ANSI, and replacement of COBOL/EMBEDDED SQL Integration features with standard Java/C# practices.

SoftwareMining CORECT changes the Embedded SQL statements from COBOL to Java / C# dialects. Additionally, the system allows externalization of SQL statements in order to ease future database migrations e.g. from DB2 to Oracle.

PERFORMANCE

Many ESQL applications are Batch-programs which need to be completed within a set time frame every day.

As such the migration of such system would hugely benefit from translation into a design which has performance as one of its main considerations.

CHANGING DIALECT OF EMBEDDED SQL

In order for the embedded SQL code to work in new system, the dialect has to change from COBOL to Java or C#. (E.g. new code should use '?' or '@' delimiters instead of *:FIELD* notation).

SoftwareMining tools achieve this automatically.

CATERING FOR DATABASE CHANGES

After the translation of COBOL to Java, it may also be desirable to change the underlying database - e.g. moving from DB2 to Oracle, or MySQL ... The main issue to overcome is the SQL return codes hard-coded in the application which differ between database servers. SoftwareMining migration process allows runtime configuration / substitution of SQL codes - minimizing efforts involved in such database changes.

MIGRATION OF COBOL STORED PROCEDURES TO JAVA OR C#

SoftwareMining tools can be used to migrate COBOL Stored Procedures to Java or C#. The new Java or C# stored-procedures can then be imported into a variety of databases such as Oracle or Microsoft SQL Server. Due to use of Object-Orientation, Exception handling, and framework based architecture, Java / C# stored procedures have better maintainability than standard Transact SQL or PL SQL scripts. The new stored procedures are run within the target database environment, leading to a better performance.

WHITE PAPERS

- [Project Managing COBOL to Java / C# migration](#)
- [Gradual migration of COBOL applications to Java: one piece at a time](#) - *reducing costs and risks.*

FINALLY

The above is a very brief summary of the experiences gained. For further information please contact www.softwaremining.com or watch [COBOL to Java/C# Project Planning and Management - 10 min Guide with Embedded SQL](#) on Youtube.