



XDM

CROSS DATABASE MOVER

All relational database management systems (RDBMS) basically function the same way. However, there are many differences, as well, that make it challenging and laborious to move data from one system to another. XDM automates this task.

Today's fast moving business environments require flexible data architecture and modeling. Ongoing change is the norm. Typically, business data is split over two or more instances of RDBMS, and it's not uncommon for some of these systems to be of different breeds. Whether it's DB2 on mainframe, or on LUW, or it's Oracle, you name it, sooner or later, every enterprise will need to move data from one type of DBMS to another for architectural or operational reasons.

For example, say you run DB2 on z/OS systems, and developers want the data on Windows or Linux machines to avoid incurring development and test workload on production LPARs. What's required just for this? If we start from scratch in this case, a completely new environment needs to be created. First of all, this means the DDL statements related to the objects involved need to be generated according to rules and syntax of the target DBMS. Each database management system has its own DDL requirements; hence the challenge is to convert/adapt the DDL drawn from the source system to a valid set of DDL statements for the target system.

Development and test environments require a data refresh from time to time. In this case the objects may already exist in the target system, but differ from the objects of the source system because developers may have altered tables since the last refresh by adding new columns or changing data types, etc. That's why a decent method for comparing the source and target objects is necessary. Appropriate measurements/treatments must be taken/specified for the detected differences before a successful data migration can begin.

These complex, and often error prone steps, are fully automated by XDM. An easy to operate workstation interface enables users to define a migration process as a copy task by specifying selection-, exclusion and rename rules. These rules are easy to specify and very flexible. For example, the rules can relate to all tables of a schema. The specs are subsequently stored under a task name in a central database, ready to be executed at any time. The task examines the source DBMS instance to select all objects that match the specified selection/exclusion rules. Next the target environment is examined and a comparison report is generated; the DDL statements to create the missing objects are built.

XDM offers flexible refresh options:

- Drop all objects of the target instance and recreate them as defined in the source
- Create only the missing objects. Incompatible objects may cause a negative return code
- Create only missing objects and skip objects that are incompatible
- Create missing objects, drop the objects that are incompatible and recreate them as they are defined in the source

In order to save machine resources and process time, XDM always chooses the fastest and most efficient copy method. For cross copies between heterogeneous DBMSs it deploys the industrial standard format CSV to transport the data from source to target. For homogenous copies between systems of the same type it uses the available native methods to directly copy the tablespaces. XDM takes care of field transformation to guarantee platform compatibility.

In a nutshell

XDM integrates and automates the entire work sequence of data migration between relational DBMSs. However the XDM migration tasks can also be split. For example, it is possible to only extract the DDL of the source objects and create an empty environment in the target system. A platform independent script language facility is available to allow creating migration tasks by programs. XDM tasks are autonomous; they are ready for unattended operation and can be executed under control of standard scheduler software.

XDM supports the most popular platforms and relational database management systems including

- DB2 on z/OS, for LUW and AIX
- Oracle for Windows, Unix and AIX
- Microsoft SQL Server

Technical Feature Summary

- Copy Task definitions with processing options, selection, exclusion and rename rules
- Central DB to host copy task definitions
- LDAP server for central authentication and permission control
- DDL Analysis of the objects of the source instance
- DDL Generation according to rules and syntax of the target DBMS
- Structure Comparison of source and target objects with generations of a comparison report
- Data Extraction from the source instance and load into the target system or direct tablespace copy (homogenous copy)
- Field transformation for platform compatibility
- Platform optimized data movement by using the most efficient technique of the respective platform
- Parallel processing
- Automatic Adjustment of max assigned values of identity columns
- Storage of intermediate data (DDL, scripts, CSV files) either on a central storage server or directly on the target database server to reduce the data transfer time
- Data Transfer using FTP or SFTP
- Scheduler controlled execution
- Platform independent batch processing language
- Autonomous migration process