



CASE STUDY | MANUFACTURING

Reinventing data migration process

with the SQL Comparison SDK

redgate
ingeniously simple

" We have a small international IT department and a very heavy workload, it was difficult to make time to thoroughly check the SQL statements to ensure they would all work correctly on the branch side. SQL Comparison SDK's checks and balances system eliminates the worry and the guesswork."

DUSTIN FRANKS

Leggett & Platt, project manager

The context

Managing accounts in more than 20 countries for customers such as Ford, Starbucks, Target, Sony and Lear is no small feat, even for a \$5.1 billion company like Leggett & Platt. To automate the task and make it less prone to mistakes, the company uses a combination of iScala, SQL Server and Red Gate software tools to manage its data processes.

Leggett & Platt designs and produces engineered components such as store displays, home furniture products and steel tubing for customers worldwide. The company's diverse operations bring numerous languages and currencies into play, and each company branch needs specific functionality from its database program. So the Leggett & Platt IT department decided to implement an enterprise resource planning (ERP) system called iScala. iScala handles international branches' day-to-day business – tasks such as sales orders, inventory tracking and manufacturing – using the native country's language and currency.

Leggett & Platt created a set of custom programs called LPScala to compliment iScala and handle internal needs such as payroll applications. LPScala also transfers general ledger information back to the company's corporate office to load into the financial planning system. Both LPScala and iScala interface with Microsoft's SQL Server, writing all of the data obtained from the programs to a SQL database.

Updates: tedious and time-consuming

Leggett & Platt has approximately 20 SQL Servers to handle the data the international facilities use to run their day-to-day business. Each server hosts around five databases. There are two trained SQL Server DBAs on hand at Leggett & Platt's corporate office in Carthage, Mo., to help with any issues that might arise.

Updating these databases across each location used to be a daunting task for the Leggett & Platt developers. They had to manually create SQL statements and bundle them into an InstallShield application. The InstallShield file was run at each location, where a Visual Basic application created the database, another executed the SQL scripts to create the database objects, and an executable file in the command line imported the data into the tables. These hand-coded scripts left plenty of opportunities for error.

Aside from reducing the chance for errors, Leggett & Platt wanted to ensure the .NET application they created for each database could be executed by someone at each branch, even those with limited SQL experience. It was also crucial that the databases and core data be updated correctly.

An accurate, repeatable process

The best solution for Leggett & Platt was to reinvent its entire data migration process. The company now uses Red Gate's SQL Comparison SDK to create a repeatable process for updating company databases. SQL Comparison SDK consists of Red Gate's SQL Compare, SQL Data Compare and SQL Packager software, together with the ability to program directly to APIs, either from the command line or natively from another application.

Franks and his group first use SQL Compare to take a snapshot of a production database, including all of the data tables. SQL Compare then identifies which tables have been changed between development and production and generates scripts to update the structure of the production database. After all of the table and schema changes have been made to the development database, tested and finalized, the database is moved to production.

For deployment at the remote sites, Franks creates an executable file using a combination of SQL Packager and InstallShield. SQL Packager creates an executable file of the final SQL database tables with data that will be distributed at the customer site, preserving the structure of the database as well as all of the dependencies within the tables. Dependencies include language entries, security items and menus Leggett & Platt establishes for user type definitions.

SQL Packager first guides users through selecting schema and data for the new database, then generates scripts to preserve dependencies. In the final step, the tool automatically packages an executable or .NET project, creating a utility for easy installation.

Franks and his team then write a .NET application for each update that will launch at the customer site. The .NET code will compare the database created by SQL Packager with the database currently in use at the Leggett & Platt branch and generate the scripts needed to update the branches database schema.

To create the final file for remote deployment, the SQL Compare snapshot, SQL Packager executable, and manually authored .NET application are compiled with InstallShield.

Support gets a boost

Once launched at the client site, the InstallShield application creates a clean database from the SQL Packager file that will be used for data synchronization. Then the .NET application compares the database snapshot from InstallShield with the data on-hand using Red Gate's SQL Data Compare, generating a script that outlines the differences in the two databases. The script is then executed to update the branch's data to match records in the SQL Packager executable, enabling the on-site databases to be updated without wiping out existing data.

Leggett & Platt's IT support for customer databases has gotten a much needed boost with the implementation of iScala and Red Gate tools. The small workforce with a heavy workload can now concentrate more on needed updates and Visual Basic code rather than double-checking scripts and creating fixes.

Erin Hatfield (info@cramco.com) is a writer for Cramblitt & Company, a marketing communications company serving the computer graphics, IT and electronics industries.

"With SQL Compare, if one of the database transactions fails, they all fail, the database is rolled back to a clean database. We used to have to repair incomplete databases with a special release or manually delete records and rerun the installation."

Dustin Franks, Leggett & Platt, project manager

Try the SQL Comparison SDK free: www.red-gate.com/sql-comparison-sdk