



February 9, 2011

## Simpler Database Migrations Have Arrived!

IBM, ANTs Software, And EnterpriseDB Offer Innovative New Options

by Noel Yuhanna

with Mike Gilpin and Adam Knoll

### EXECUTIVE SUMMARY

Database migrations have always been complex, time-consuming, and costly due to proprietary data structures and data types, SQL extensions, and procedural languages. It can take up to several months to migrate a database, depending on database size, complexity, and usage of these proprietary features. A new technology has recently emerged for solving this problem: the *database compatibility layer*, a database access layer that supports another database management system's (DBMS's) proprietary extensions natively, allowing existing applications to access the new database transparently. ANTs Software, EnterpriseDB, and IBM offer database compatibility layers that enable a new approach to database migrations. Application developers and database administrators looking to migrate databases due to cost concerns, staffing problems, or technology issues should consider this new database compatibility layer option to mitigate risk, lower migration cost, and speed the migration process.

### DATABASE MIGRATIONS HAVE ALWAYS BEEN CHALLENGING AND RISKY

Database migrations have always been complex, time-consuming, and expensive, because each DBMS product has its own proprietary data structures, data types, and SQL extensions. Once a firm commits to a DBMS, applications it builds on that platform tend to use those proprietary features, creating a major challenge when the firm tries to migrate to another DBMS. Firms Forrester interviewed said it typically takes six to nine months to migrate a large database application — and in some cases years, especially when the migration requires an application rewrite. Traditional migration tools and services can help, especially with data and schema structure migration. However, these solutions don't offer comprehensive support for application code compatibility, which means that they still require costly rewrites.

“We didn't want to spend a million dollars on database migration just to save a million dollars on a new DBMS platform; it just didn't make sense. As a result, we chose not to migrate, but if there were a simpler approach, we would move right away.” (IT manager, North American retailer)

Enterprises migrate databases to:

- **Lower database cost.** Cost savings on database licenses and support are the primary factor motivating enterprises to migrate to another DBMS product. Left unchecked, database cost can become a significant part of the IT budget, and with the growing demand for database deployments, enterprises are looking for ways to lower these costs, including DBMS migration.



#### Headquarters

Forrester Research, Inc., 400 Technology Square, Cambridge, MA 02139 USA  
Tel: +1 617.613.6000 • Fax: +1 617.613.5000 • [www.forrester.com](http://www.forrester.com)

- **Address staffing shortages.** Unlike firms that are employing leading DBMS products such as IBM DB2, Microsoft SQL Server, and Oracle, some firms — especially those in smaller cities or more-remote locations — are encountering shortages in expert staff for their DBMS product. Each year, a few enterprises migrate their DBMS to a leading DBMS largely because they are having difficulty finding qualified developers and database administrators (DBAs).
- **Increase performance and scalability.** Some organizations migrate mainly because their current DBMS has performance and scalability limitations; this is especially a problem for firms employing very large (multiterabyte-sized) databases or supporting hundreds or thousands of users.
- **Rationalize database standards.** More enterprises are standardizing on as few as two DBMS products to minimize costs and improve operational efficiency. Standardization often triggers a database migration initiative.

## DATABASE COMPATIBILITY LAYERS BRING A NEW APPROACH TO DATABASE MIGRATION

Firms traditionally accomplished database migrations in three phases: migrating the schema, migrating the data, and migrating the application code. Although tools have been available to automate schema and data conversions, application code changes remained a major challenge. The database compatibility layer (DCL) offers a new solution to this problem that speeds migrations to a matter of days or weeks by enabling existing applications to work against your new DBMS with only minimal changes. Although a DCL does not support 100% application code compatibility, firms Forrester interviewed indicated that it's possible to achieve 90% to 98% compatibility, which is a significant improvement over traditional approaches. Forrester estimates that more than 400 companies have migrated their databases using a DCL, and we expect this rate to double in the next two years.

A database compatibility layer is:

*A database layer that supports another DBMS's proprietary SQL extensions, data types, and data structures natively. Existing applications can transparently access the newly migrated database with zero or minimal changes.*

Currently, IBM DB2 is the only DBMS product that offers a DCL, but more are likely to follow suit. IBM offers DCL support with DB2 for Oracle and Sybase databases, including support for PL/SQL and T-SQL, respectively. IBM jointly developed DCL technology with EnterpriseDB for Oracle compatibility and ANTs Software for Sybase ASE compatibility. Both ANTs Software and EnterpriseDB offer their own migration tools to support data, schema, and application migrations. The DB2 DCL allows application SQL code (queries, functions, triggers, and stored procedures) written for Oracle or Sybase to run natively against the DB2 DBMS with zero or minimal application changes. Since IBM integrated its DCL natively into the DB2 DBMS kernel, it does not perform any code conversion but instead executes the application code natively as if it were written to DB2.

One customer pointed out an application performance improvement upon migrating to DB2 from Oracle using the DCL — a migration that required no application code changes. A DCL can help:

- **Accelerate database migrations.** A DCL significantly reduces the migration effort compared with traditional database migrations that require significant data, schema, and application code changes. Based on customer feedback, expect that a DCL can reduce migration effort by more than 70% compared with traditional database migration approaches.
- **Mitigate risks.** With a DCL, application changes are minimal; when combined with automated data and schema conversions, a DCL significantly reduces the risk of introducing new bugs that result from having to modify the code. Traditional approaches require much more application change, which requires much more testing to verify that the code still works correctly.
- **Lower migration cost.** Database migrations can be costly, especially for large and complex applications. DCL automates the entire process of data, schema, and application migration, requiring much less effort and cost.
- **Preserve the relevance of developers' skills.** A DCL does not require developers to learn a new programming language or DBMS application program interface (API). For example, when using a DCL, developers can still code in Oracle PL/SQL language to access an IBM DB2 database.

## RECOMMENDATIONS

### A DCL IS THE MOST COST EFFECTIVE AND LEAST RISKY MEANS OF DATABASE MIGRATION

The new economics of using a DCL to effect a database migration mean that you are no longer as locked in to a DBMS product as you were before. A DCL offers a new and viable option for database migrations that you can use for any application, big or small. If you are considering a database migration and plan to use a DCL, consider:

- **Performing planning and assessment before any migration.** Don't forget the planning and assessment phases in database migrations, which are as important as ever, even when using a DCL. Assess the viability of the application for migration, and involve the appropriate stakeholders, including the business owner, the data architect, the DBA, application developers, and business analysts.
- **Freezing application code changes during the migration process.** Freezing other changes to application code throughout the migration process can dramatically minimize application issues. With traditional database migration approaches that took several months to complete, this would have been impractical. However, with a DCL that can perform migrations in days or weeks, freezing application changes is much more likely to be feasible.

- **Testing the application as you normally would.** As the majority of application code will not change, using a DCL should reduce the need for *deep* testing of the application; however, testing is still critical. In addition to normal application testing, conduct performance and scalability testing of the application and database to ensure that the new levels meet your requirements.
- **Upgrading hardware in parallel when possible.** When migrating a database, also consider upgrading your hardware, especially if an upgrade is overdue or if you will need to support a larger database or more users in the near future. Using a DCL for database migrations typically has no impact on hardware upgrades, especially if you are migrating between various flavors of UNIX systems — such as AIX, HP-UX, Linux, or Oracle Solaris — and you can do the upgrade in parallel.