

20 September 2002

File: SIS 1017

Leveraging Native DBMS ETL Tools

Server Infrastructure Strategies

Mark Shainman

Given growing data volumes and data sources, data warehouse project teams should evaluate the native functionality of their databases to cost-effectively meet extract-transform-load (ETL) needs.

Collecting business transactional data within a centralized warehouse environment is critical for driving business processes ranging from ERP to CRM applications. Consequently, the process of effectively extracting the data from the operational systems, transforming it, and loading it into the decision support system becomes not only the most important process in constructing an effective decision support infrastructure, but also the most complex and potentially the most expensive. Our research indicates that the most extensive ETL processing is still accomplished through homegrown programs. The cost of maintaining this code is becoming increasingly problematic, yet the cost (e.g., licensing, training, additional coding) of third-party ETL tools (e.g., Informatica, Ascential) remains high as well. As IT organizations (ITOs) look to decrease costs by leveraging the native ETL functionality of their databases (i.e., Oracle, Microsoft, and IBM), three questions must be addressed:

- What are the target databases?
- How many different types of data sources does the organization have?
- How complex is the process?

During 2002-04, database vendors will concentrate on expanding preset transformation functionality as well as expanding source access. Through 2004-06, users can expect DBMS vendor ETL products to continue to target only their own databases, while heterogeneous source access will gain parity with third-party ETL vendor products.

What Are the Target Databases?

Database vendors continue to integrate ETL into the database engine as a means to make it easier for each vendor to target its particular database and, of course, expand its usage within the organization. For example, Microsoft offers Data Transformation Services (DTS), Oracle offers Warehouse Builder (OWB), and IBM offers Warehouse Manager (WM). End users seeking an ETL tool that is target-database-“agnostic” must look to third parties.

Both Oracle’s OWB (see Figure 1) and IBM’s WM (see Figure 2) use the transformation abilities of the target database itself, utilizing stored procedures to facilitate the transformation process and manage metadata, which makes a heterogeneous target environment impractical.

SQL Server DTS (see Figure 3) has more flexibility operating in a heterogeneous target environment. Unlike the solutions from Oracle and IBM, Microsoft’s solution should be viewed more as an ETL development platform than a true ETL tool. DTS is a platform in which users can build the transformation components, which reside outside of the database, using ActiveX scripts and creating COM objects in C++, which, like any homegrown ETL tool, can target not only the SQL Server platform but also other databases. Although other platforms can be targeted, all preset transformations and load capabilities in DTS are geared toward the SQL Server platform, and most organizations also leverage Transact-SQL within the database for part of their transformation process.

META Trend: By 2005/06, Intel-based servers will dominate at the DBMS tier. Scale-out database configurations will grow in popularity as management virtualization improves, but remain primarily a high-availability option. By 2004, XML-based content storage and rendering will become a key DBMS differentiator along with data movement and transformation. Centralized data warehouse architectures will emphasize database workload prioritization and data archiving to support the increasing need for near-real-time analytics through 2006.

What's the Source?

The number, type, and complexity of extraction sources (e.g., mainframe flat files, applications, message queues) are key to determining whether embedded ETL tools can be leveraged. Unlike third-party tools, tools from Oracle, IBM, and Microsoft do not have native access to most data sources, but the three companies support access to all relational sources — through ODBC or gateways (see Figure 4). As a result, extraction performance is impacted. Although improvements have been made in ODBC performance, and native ETL tool source access, organizations that have highly complex data source environments and high throughput requirements must look to specialized third-party ETL tools to provide the largest selection of data source access. Through 2004-06, ITOs can expect Oracle to lead the way in native source access, with IBM and Microsoft following. Improvements in ODBC driver performance will enable all three native tools to extract from a greater number of data sources, with near native performance.

How Complex Is the Process?

Complexity of data transformation, volume of data, and data freshness requirements are also important attributes that must be defined. ITOs requiring highly complex transformations and large data volumes — often the case when a separate server is needed to facilitate the transformation process — will find it more cost-effective to go with a third-party ETL tool (see Figure 5). In dealing with complexity, our research indicates that all three products — OWB, WM, and DTS — can accomplish complex transformations, but the amount of work needed by each of these tools to set up the transformation process differs greatly:

- Out of the three tools, Oracle's OWB has the advantage of being able to handle, out of the box, highly complex processes made up of complex data transformations and large data volumes. ITOs can benefit from recent advancements in Oracle9i's SQL functionality and preset transformations, which enable end users to much more easily set up and execute the transformation process (see Figure 6).
- DB2 WM, like OWB, can leverage the power of SQL within the database to implement the transformation and loading of the warehouse. It has the ability to accomplish complex transformations, but unlike Oracle, which can leverage PL/SQL as a stored procedure language, WM must rely on the capabilities of ANSI standard SQL, Java, or C++ to implement the transformations, making the transformation setup process more complex.
- Microsoft's DTS has the most limited out-of-the-box transformation abilities of the three native ETL products. Complex transformations can be accomplished using ActiveX or by creating a COM object in Microsoft Visual C++, but organizations seeking to leverage DTS within a complex transformation environment must be prepared to write most of the code, just as with homegrown ETL programs, to facilitate the transformation.

Ultimately, as the diversity of data sources and targets increases, organizations seeking adaptability and flexibility within their ETL infrastructure should look to third-party tools. Organizations that have traditionally leveraged homegrown ETL tools and wish to leverage their existing database investment can benefit from the added functionality and preset transformations of the native DBMS ETL tools. Although these products can be considered a cost-effective way to populate vendors' databases, they are rigid and limited in their scope as overall ETL solutions. Small to medium-size enterprises with homogeneous environments are much more likely to be able to leverage native ETL functionality as a cost-effective method of achieving the ETL process.

Bottom Line

IT organizations looking for infrastructure adaptability and flexibility ahead of cost should not put forth the effort to leverage native DBMS extract-transform-load tools. However, enterprises that have tight data warehousing budgets and are finding the spaghetti-code-generating, hand-coding approach to ETL unacceptable should consider leveraging native DBMS ETL alternatives.

Business Impact: Implementation of an agile information architecture within complex enterprises requires an independent and open architecture for data movement.

Figure 1 — Oracle9i: Oracle Warehouse Builder

As Oracle touts the decision to support integration of its 9i product set, Oracle Warehouse Builder (OWB) has become a key component of its warehousing platform and is included in the Oracle9i Developer Suite. OWB is a PL/SQL code generator that leverages the Oracle database's SQL capabilities to implement the data transformation and loading process. OWB also acts as the mechanism to manage metadata, which is in turn stored in Oracle tables. Recent advancements in SQL and PL/SQL, such as Upsert functionality (insert and update), external tables, and table functions have improved OWB's ETL functionality within homogenous Oracle target environments. OWB contains more than 150 preset transformations and is also able to create user-defined transformations and store them in a function library for reuse. In addition, users can leverage OWB's scheduling engine to schedule specific tasks within the ETL process.

Source: META Group

Figure 2 — IBM DB2 7.2: Warehouse Manager

IBM's DB2 Warehouse Manager (WM) is a platform that generates code to leverage the power of SQL within the database to implement transformation and loading of the warehouse. DB2 WM uses the capabilities of ANSI standard SQL, Java, or C to substantiate the transformations. DB2 includes the concept of table functions, which end users can leverage to source from OLE DB, MQ, and flat files.

Unlike Oracle, IBM positions WM's ETL functionality as a product for organizations to leverage within medium-size IBM homogenous environments. IBM recommends and resells both the Ascential DataStage product and the ETI*Extract® product for organizations looking to meet their DB2 warehouse needs within a heterogeneous environment. IBM does tout the use of its separate federated access product Data Joiner, as a way to access and target heterogenic data sources, but our research indicates that most organizations choose to use an integrated third-party ETL tool in these situations, attempting to leverage Data Joiner with Warehouse Manager.

End users leveraging DB2 WM can benefit from extractions taking place within the IBM family of products. DB2 can natively access the CLI (Call Level Interface) of DB2 on the mainframe. IT organizations can use WM to access data from applications such as SAP and i2, but there is an extra cost per module, on top of the relatively expensive \$10,000 per processor for Warehouse Manager.

Source: META Group

Figure 3 — Microsoft's SQL Server Data Transformation Services

IT organizations should seek to leverage Microsoft's SQL Server 2000 DTS (Data Transformation Services) more as an ETL development platform than as a true ETL tool. DTS natively supports only 10 prestored transformations and has little native extraction or transformation functionality. Unlike Oracle's OWB and IBM's DB2 WM, Microsoft DTS does not leverage stored procedures within the database to perform the transformation process. It is a platform in which users can build the transformation components using ActiveX or creating a COM object in Microsoft Visual C++ — end users have the option of leveraging Transact-SQL within the database as a transformation component, but DTS will not help users with the process.

End users cannot look to DTS to natively source from any application or database except SQL Server. DTS contains the ability to source from any OLE DB or ODBC data source that meets ODBC compliance. IT organizations must leverage ODBC adapters from companies such as SAS to access applications, message queues, etc. Users cannot look to DTS to provide scheduling engine functionality, but must leverage NT, or SQL Server, which can provide a time-based or event-based scheduler that can trigger a SQL Server job.

Source: META Group

Figure 4 — Source Access

- Oracle and IBM can access mainframe sources. Oracle acquired COBOL code generators from Carlton, which gives access to the mainframe (e.g., VSAM, IMS, DB2), but this capability has been slow to be adopted. IBM, on the other hand, can natively access the CLI of DB2 on the mainframe, but must use ODBC or gateways to access any other data sources. Both vendors offer add-on modules to access applications such as Peoplesoft and SAP, but high costs for these modules are “extra” and performance is questionable.
- Microsoft’s DTS has no native data source access, but like the Oracle and IBM ETL products, it can source from any OLE DB or ODBC data source that meets ODBC compliance.

Source: META Group

Figure 5 — ETL Tool Costs

- Oracle Warehouse Builder cost is included within the Oracle Developer suite and is priced at \$5,000 per user. OWB must be used in conjunction with an Oracle database instance. If the ETL process is given its own server, the cost would be the licensing cost of the Oracle database on that server (\$40,000 per processor), plus the cost of OWB. When put on a box that has three or more processors, this is more expensive than a product such as Informatica (which has the pricing scheme of \$93.50 for 1 processor; \$121 for 2-4 processors; \$160 for 5-10 processors; \$200 for 11-20 processors; \$270 for 21-44 processors; and \$384 for 45-64 processors).
- IBM’s Warehouse Manager costs \$10,000 per CPU, but also contains IBM’s Query Patroller product. WM, like OWB, must be used in conjunction with a DB2UDB database instance. If a separate database on a separate box must be leveraged in a complex transformation environment, the cost of DB2 is \$20,000 per processor, plus the cost of WM. When put on separate server that has five or more processors, Warehouse Manager is more expensive than a product such as Informatica.
- There is no extra cost for Microsoft’s DTS, just a fee associated with the SQL Server database that it is targeting. DTS, like OWB and WM, must always reside on top of a SQL Server instance. The database cost is \$20,000 per CPU for Enterprise Edition, or \$5,000 per processor for Standard Edition. Organizations also have the option of going with a per-seat licensing model priced at \$11,099 for 25 client access licenses (CALs — Enterprise Edition).

Source: META Group

Figure 6 — New Oracle9i Functionality

Oracle has added new SQL functionality such as external tables (also available in DB2) and table functions to improve the ETL process. External tables give users the ability to map external objects (i.e., transfer flat files to tables and manipulate them through the use of SQL as if they were tables). The advantage of using external tables is that, through the use of SQL, data can be loaded into the database and joined to another table in one step to accomplish tasks such as key validation. When users create external tables, they create a mapping between the object, and the mapping resides in the data dictionary.

Table functions are new type of function within the Oracle database. PL/SQL used to appear in the “Select” function. With table functions, it now can appear in the “From” function, which enables input and output to each be a stream of rows. Previously in PL/SQL, an input and an output were each a single row.

Source: META Group