APPLICATIONS

# Data Virtualization: An End-to-End Process Approach

By ROBERT EVE

**ENORMOUS DATA VOLUMES** in complex systems exacting high total cost of ownership (TCO) are endemic in today's enterprises. Must this always be the case? Not for enterprises and agencies using today's advanced data virtualization to simplify data complexity and reduce costs, time to solution and risk.

Today's data virtualization is used at development and run time for integrating data from multiple, disparate sources—anywhere across the extended enterprise—for consumption by a broad range of front-end business solutions. It is deployed on a project basis as a complement to other data integration approaches. It is also implemented as a complete architecture as a data services layer in service-oriented architecture (SOA) environments. This article covers project-level cases only.

## THE DATA VIRTUALIZATION LIFE CYCLE

Compared to other popular data integration methods (e.g., consolidation and synchronization), data virtualization is faster for both first-time solution development and ongoing improvements. Enterprises wishing to achieve best-practice results typically use the following six process steps: 1) gather requirements, 2) discover data, 3) model data, 4) build solutions, 5) operate, and 6) improve.

## GATHER REQUIREMENTS

Accurately translating information needs into technical requirements for data integration and application function requires time and thoughtful effort. Commonly, this task proceeds in a linear fashion, usually in a conference room, with the results captured in a document. Unfortunately, significant gaps often arise between real requirements and the delivered solution. The inexactness of human language makes it difficult to accurately capture requirements in isolation. Furthermore, implementations are often long-term, so the requirements frequently change before the solution is finally delivered.

On the other hand, data virtualization's life cycle is iterative and interactive, enabling rapid visualization of prototypes based on real data. Analysts can quickly show lightweight (but real) versions of the information to measure against business users' requirements. These can be quickly modified as necessary. Because the entire end-to-end process is shorter, fewer business information need changes typically occur before deployment.

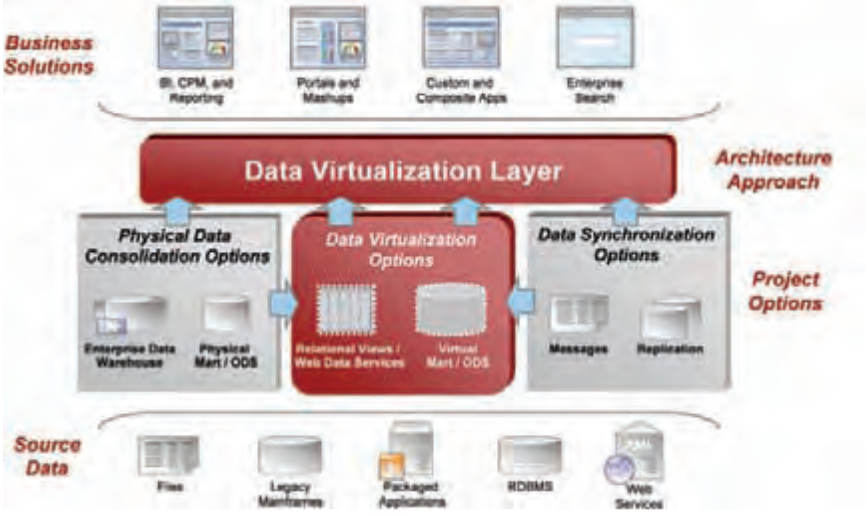


Figure 1: Data Virtualization at a Glance

## DISCOVER DATA

After the data requirements are understood, finding and assembling specific data from the vast enterprise information system can be challenging for a variety of reasons. For one, developers are unlikely to be familiar with every potential source. When a global computer manufacturer wanted a summarized inventory across its Asia-Pacific manufacturing pipeline, in-house developers were unaware of every supply chain data source. In contrast, data virtualization's discovery capability automated the identification of potential data sources.

Sometimes, there are regulatory compliance requirements legally mandating that 100% of the data be verified. When a global investment bank needed to find missing trade information from multiple trading databases, it spent weeks of manual effort to ensure the legal levels of completeness. Data virtualization's discovery capability automated the search, reducing the length of time required.

Often the same data is replicated (and therefore redundant) in multiple sources. To determine the best data source normally requires manual inspection and ad-hoc querying. For example, information about a customer may be found in multiple systems, including order management, invoicing, collection and customer service. Data virtualization's discovery capability automatically points to replicated data, making it faster to determine the best source.

# APPLICATIONS

## MODEL DATA

After locating the correct data, developers must find relevant relationships, or how data element "X" relates to data element "Y." The data and its relationships form a model, and it is important to understand this model to properly construct meaningful solutions. This can be challenging for two reasons. Within a single data source, the schema may be documented, but frequently the documentation is out-of-date. Additionally, relationships between data in different data sources are rarely documented. Yet these inherent, cross-database relationships are necessary to combine data into something meaningful.

Modern data virtualization platforms provide three helpful capabilities for identifying data relationships. The first is a set of prebuilt models for most popular packaged applications, such as SAP, Siebel, Oracle Apps, PeopleSoft, etc. These models eliminate the need to figure out complex and obtuse models, therefore allowing developers to focus on the unique models specific to their enterprises.

The second is relationship discovery that automatically finds and reveals relationships inherent in the data. These are accomplished using advanced data profiling and indexing techniques. Developers use the discovered relationships to assemble complete data sets to meet the requirements.

The third is a transfer capability that transfers completed data models to the data virtualization server for current and future use-cases.

## BUILD SOLUTIONS

The next process step is to build a set of tabular views that can be queried by traditional BI and reporting clients, or data services accessible by application and mashup consumers in an SOA. (See Figure 2.) Many data virtualization deployments will offer the same data set via both mechanisms, allowing for reuse and consistency.

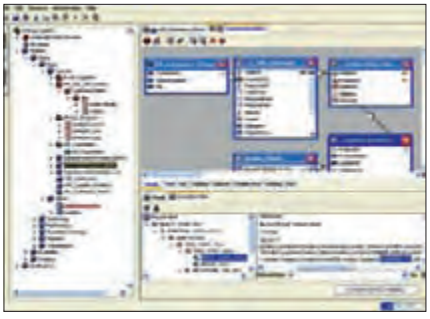

**Figure 2: Studio for Developing SQL-Based Views**

Modern data virtualization platforms provide an advanced development environment for building both tabular views and data services. These drag-and-drop graphical tools eliminate most hand-coding to enable faster development by a wider range of skill levels among developers. Once the views and data services are available, developers can focus on building the consuming application. By partitioning the data integration from the application logic, developers can build simpler applications, and different developers can work on different parts of the problem concurrently. Further, the partitioning promotes reuse because existing views or data services may satisfy new data requirements that arise.

## OPERATE

Once a solution has been developed, the focus turns to the runtime or operation phase of the life cycle. The operation phase differs from the development phase. Operation is all about performance, scalability and reliability. With data virtualization, data remains in its native form in the system of record. This has numerous advantages, but it may introduce the new challenge of maintaining sufficient performance and scalability. Several advanced strategies to maintain optimal performance include advanced federated query optimization to speed performance; explicit and implicit caching to temporarily stage data for repeated use, saving cycles; and multiple-server clustering to increase capacity and improve reliability.

## IMPROVE

Because applications are more loosely coupled and rampant data replication has been curtailed, improving data consistency and increasing governance is easier to achieve. Modern data virtualization platforms offer mechanisms for easily maintaining the models, views and services. These help form a buffer-zone abstraction layer between the data sources and data consumers that absorb and accommodate changes, thereby enabling continuous improvement.

## SUMMARY

Enterprises and agencies are realizing a range of benefits from modern data virtualization platforms because these platforms offer quicker access to complete data; more accurate data modeling in less time; and reduced complexity. Data virtualization typically lowers the costs for development, implementation and maintenance. Finally, data virtualization enables continuous improvement because the process is easier to change as information needs evolve. ■

**ROBERT "BOB" EVE** has held executive-level roles at several leading enterprise software companies prior to joining Composite Software, Inc., where he is executive vice president of marketing. He is a frequent speaker and contributor to publications including *DBTA*. For more information about Composite Software, go to www.compositesw.com.