

# Data Integration

## Executive Summary

Lacking proper tools, the vexing challenges relating to bringing two organizations together deadly serious business for an entire corpus of employees, owners, and clients. Merging implies the need to acquire the best tools available, and compromise will bode evil in the end. The merger paradigm is yet a durable example for in-house systems integration and application extension and refactoring.

Data Integration, being the first order of business for merging systems, is an issue often lost in the planning process. The machines that have quietly churned away in the back-room are generally reliable, invisible, and easy to forget. However, the closer we get to defining post merger IT roles, data integration and new processes creation brings the issue quickly to the fore.

We need power tools and a data serving infrastructure that will not fail us. The monumental journey towards a unified data model must be as robust and non-disruptive as possible – excuses will not be tolerated.

OpenLink Software's Virtuoso Universal Server is such a power platform for integration of data using the flexible and powerful methods of Virtual Database technology and Programmatic Transactional Replication. Virtuoso encompasses OpenLink's 12 years of innovation in Universal Data Access, ODBC standards, and reliable replication technology.

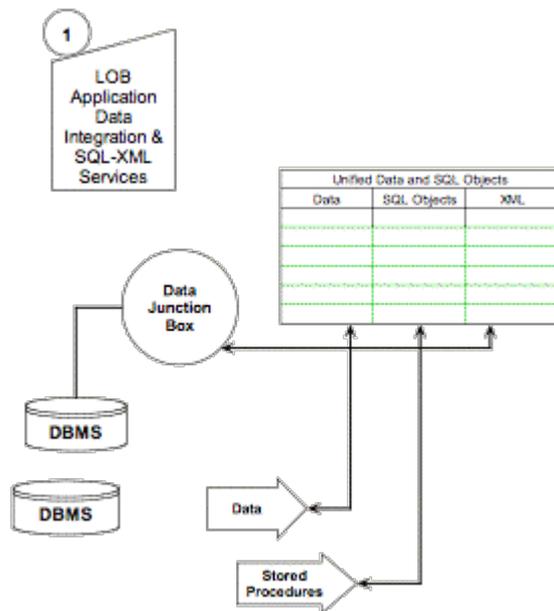
## The Unified Data Model for Post Merger Capital Line Applications

The contemporary mantra of 'A-List' Database vendors is the creation of the 'single data model'. After years of campaigning to keep licensed sites and users from ever migrating out and connecting to other data sources, these giants of industry thought have finally come around to the realization that applications proliferate, companies merge, and on-line partner-based business models connect.

The Internet's open model of distributed data was a renaissance for many innovative startups; smaller, agile, and more innovative companies, like OpenLink, pioneered the

open data connectivity model that the behemoths now espouse. OpenLink's tenure in Universal Data Access architecture is the fertile ground that gave root to the Virtual Database technology in **Virtuoso Universal Server**.

In the Unified Data Model, a heterogeneous collection of databases is transparently represented as a single logical unit. In an IT driven merger scenario, line applications are now able to access all data storage through a single control point, or 'data junction box'. More significantly, the unified model channels application requests via a single SQL dialect, provides unified administrative access, security, and the stability of keeping existing systems intact. This is the magic of the Virtual Database.



Since both client API and SQL dialect are normalized by a virtual database, the inevitable post-merger divergence of the inherited Database Systems are rendered eminently manageable. In a single, mighty stroke, the CIO's worst nightmares are transformed.

A proper virtual database must offer a comprehensive set of client API's in order to preserve connections to existing applications[1]. Virtuoso offers just such a complete solution, adapting to existing environments, while preserving the investment in application logic, stored procedures, and database design.

The unified data model provided by Virtuoso spells relief in the merger me'lange, so let us count the ways:

1) Transparent distributed querying capabilities, hiding both locations of data as well as the limitations of the system hosting the data. The entire disparate infrastructure becomes accessible through a single set of API's, covering all major standards,

ODBC, OLE/DB, JDBC and .net.

2) Time. By unifying data access and preserving the attached systems, harried system administrators and IT analysts can summon a little breathing room while contemplating larger system issues, and the inevitable introduction of new systems and modern technologies, such as Web Services.

3) Virtuoso provides a path to web services capabilities for all attached data sources, creating an ideal gateway for bridging existing line application functionality between the merged systems *and external trading partners*. SOA, or Service Oriented Architecture, can now become a high value benefit of a merger's former potential for disaster. Virtuoso also provides complete XML handling and transformation functions, making the Web 2.0 and e-commerce transition possible.

4) The foregoing is incremental, requiring no re-engineering of existing processes. Installation is easy to accomplish through a web-based interface, allowing attachment of remote data sources and user account configuration. Existing applications and databases remain intact.

The Unified Data Model provided by Virtuoso Universal Server ties up the loose ends of many systems being forced by events to work together. Existing applications and systems can be preserved, kept in place, and migration deferred until the post-merger dust is considered sufficiently settled. For a more technical and detailed exposition of Virtuoso's Virtual Database, see [\(link here\)](#).

It is an elegant solution, provided in a robust and simple package – yet there is another strategy for shared data availability that may also apply in certain situations where full-scale unified access may need to be deferred – *Programmatic or Transactional Replication*.

## Transactional Replication for Keeping Merged Systems in Sync

The Virtual Database mentioned previously is the best way to create a single data model for a diverse pool of systems. While replication does occur in these system, it is usually based on duplicating transaction data – not a replication event. Virtuoso can push data to any number of unified systems, however the single data model should be viewed as a mode of operation, in most cases of data integration, the primary mode.

Transactional Replication is a practical integration solution in well defined circumstances. Replication can provide low-impact data availability and reliability, especially for systems that are intermittently connected or, for partner systems that do not need direct application access to a central database.

Our two replication techniques are as follows:

- The Virtual Database allows maintaining consistent copies of SQL data in a network of heterogeneous servers. Most databases can be bi-directionally synchronized, and data maintained in Virtuoso can be periodically pushed via update triggers on supported remote databases, including Oracle, MS SQL Server, DB2 and others.
- Transactional Replication - is used for keeping relational and object repository data in tight synchrony between Virtuoso servers. Data replication flow can be one way from a central publisher to a number of subscribers, two-way with conflict resolution. Transactional replication is supported at the database engine level. This mode of replication is available in two variants, two-way with conflict resolution and one-way from publisher to subscribers.

Replication is more than viable when one partner has primary custody of the data, and where the application servicing infrastructure is kept intact. A transactional replication between the inventory server of Company 'A' can be published to Company 'B' with no changes in operations, IT systems, or training methods. The elegance of the technique is that the published data is fully usable as a transaction source, and may be updated locally.

Publish/Subscribe model replication is efficient in terms of response time and customizability.

The basic unit of transactional replication is the publication. Transaction changes to the master database are recorded in a publication log. The publication log contains the history of an upcoming publication instance, and is replayed on subscribers, similar to a recovery log. Each transaction is serialized with a transaction log number, and each subscriber is reconciled to this transaction sequence number.

In this way, transactions are received in order as they are committed, and only whole transactions are ever received on subscribers. The subscribers need not be continuously connected, although they certainly may be. Transactional replication supports logging arbitrary procedure calls into the publication log, with the result of logical operation being transferred to target subscribers. This offers possibilities for integrating application intelligence into the replication.

Transactional replication is well suited for load balancing - an common issue in departmental mergers. While not offering the same atomic consistency as a two phase commit cycle (as in a Virtual or native database), it provides a reliable alternative, as the subscriber is commonly only milliseconds away from its publisher, and will catch-up on subsequent transactions. Finally, a publisher alone decides whether a transaction is committable. After a subscriber has caught up with the publisher, it stays connected, receiving the feed of fresh transactions as soon as they are fully committed.

**Virtuoso Universal Server** is an ideal central replication manager. As a front end to

other databases, Virtuoso can control updates based on data from multiple replication feeds. Event triggers on remote database servers can be marshaled to Virtuoso for inclusion into a transactional publication. With little programming, Virtuoso can be used as replication controller front-end for linking dissimilar databases into a transactional relationship. Unlike the unified data model, in the VDB case, the replication does not require a copy of the working tables involved.

## Conclusion

Unified data model via the Virtual Database, or transactional replication<sup>1</sup>

**Virtuoso Universal Server** offers both methods and the ability to change without penalty. For a CIO facing crucial system-wide decisions, Virtuoso is a most flexible power platform for effectively managing IT system time and resources during the course of reengineering.

The unified data model is apropos when two-become-one. If your IT departments are intended for a full meshing and migration to the web services model, the **Virtuoso VDB** is a data junction box without peer. Going the VDB route brings unity out of diversity, and opens the way to advanced web services and XML based SOA.

Transactional Replication, with Virtuoso in multiple instances or mediating as a Replication Controller, is appropriate when one department may be serving the primary load for a given application set, or each partner decides, for a time, to keep current systems and applications as they are. In this common scenario, Virtuoso Replication services can insure that data is mirrored and available on-line at both partner/department sites.

## Learn More

- [Virtuoso Documentation](#)
- [Virtuoso Tutorials](#)

[1] ODBC, JDBC, OLE/DB and .net data adapters