Market Requirements Document

Feature Name: Relational Database Gateway

Version: 1

Completed By: Leon Guzenda

Description of the Problem

Many of the large projects that evaluate Objectivity/DB have a requirement to read, incorporate or update data held in existing (a.k.a. legacy or heritage) databases, as well as Objectivity/DB. The existing databases are mainly RDBMSs, with Oracle and DB2 being predominant.

Description of the Requested Feature

A number of RDBMSs have incorporated server-side gateways that can access other RDBMSs. For instance, Oracle bought Toplink and it is now available as a server side option. Queries directed to an Oracle server may return data that has been obtained from an external DBMS. This loosely fits the definition of a true federated database. IBM offers a federated database product (DiscoveryLink) that works with DB2 and other databases (and some structured files) to return data from multiple sources.

Objectivity/DB was architected to be a federated database, with the ability to incorporate external databases within a federation. However, we opted not to build the capability in earlier versions, mainly for logistical reasons. There is now sufficient demand, particularly from Homeland Security and the Intelligence Community, to offer an optional relational database gateway.

There must be ways to:

- Define the mapping between Objectivity object definitions and external RDBMS dictionaries (which define tables and columns).
- Bind an external database connection mechanism to an instance of an Objectivity database.
- Apply regular Objectivity methods (including iterators and queries) to "objects" that are actually rows (real or via a VIEW) in an external RDBMS. RDBMSs do not generally support updates via a VIEW, so this would acceptably restrict some methods, such as updates.
- Support associations between true Objectivity objects and external rows.
- Support indices, maps and other collection classes that include true objects and external rows.
- Support a reasonable subset of Active Schema functionality for external "objects".

- Support a reasonable subset of our DBA tool functionality. For instance, a request to create, move or delete an external database would be denied. A request to perform a backup should be acceptable and should produce consistent backups.

Part of an existing feature or does it require another feature, if so, which one?

The feature would be offered as an option.

How is this problem being solved now, and why isn't that acceptable?

It has always been possible for an application writer to access both Objectivity/DB and external databases. However, until we incorporated J2EE the synchronization of updates across DBMSs was unsafe. Administration, such as synchronizing backups, is a problem if there is no collaboration between the products. The application writer also has to deal with two or more database interfaces, which is tedious and error prone, particularly when inheritance and cross database relationships are involved.

What languages must support this capability?

- C++
- Java.
- Smalltalk
- SQL++

Which platforms must be supported?

- At least tier1 and tier 2 platforms, but we should develop this capability on an opportunistic basis.

Do any competitors already have this feature?

- Versant used to offer a gateway to RDBMSs, but they withdrew it when the supplier of the underlying technology (UniSQL) started to compete and then went out of business.
- ObjectStore offered a product from Dharma Software (the same one we used) for a while.
- Oracle offers TopLink (RDBMSs only).
- IBM offers DiscoveryLink.
- GemStone had a relational gateway.

Customers/Prospects who require this feature

- Homeland Security
- NSA
- CIA
- CUNA Mutual

Revenue at risk, or which could be won

- We might have been a real contender for the CIA Quantum Leap program if we had already incorporated this technology.
- We might have been more attractive to the Trailblazer project.
- Each project is potentially at least a $0.5M opportunity.

When is this required?

- Release 10.

Additional Notes

1. Third Party Solutions may be used to help implement this functionality. We have looked at products from Persistence, Dharma and Kabira in the past. None of them were entirely suitable. We are currently investigating a very promising product named JDX, from Software Tree. Commercial arrangements for incorporating and supporting Third Party Solutions must be agreed before proceeding to a detailed Requirements Definition.

2. It will not be necessary to support database (or container) replication for external databases. However, it may be viable to support some form of the other High Availability features.

3. We will also need:

- Marketing collateral

- Sales training material

- A list of publications and industry events to be approached/attended

- Training material (traditional and web-based).