Market Requirements Document

Feature Name: Resilient Networking

Version: 1 **Date:** 08/06/07

Submitted By: Nic Caine and Leon Guzenda

Description of the Problem

Objectivity/DB is architected to run as a distributed system that relies upon a network that provides medium bandwidth (100 Mbits or more), low latency (a few percent of the cost of a disk read or write operation) and reliable connections. If the bandwidth is too low, or the latency is too high or the network connections are unreliable then performance suffers and transactions may fail.

Description of the Requested Feature

This enhancement will improve client to server communications mechanisms to provide:

- The ability to operate in traditional client-server mode over low bandwidth networks.
- Improved performance in high latency networks.
- Better resilience in unreliable networks.

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

- Adds a message based Application Programming Interface (API) for C# and Java.
- Enhances the kernel's TCP/IP communications layer.

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

Application developers have to use their own messaging mechanism between clients and servers operating over low bandwidth networks. They have to build their own servers or use an applications server. Although the latter is increasingly popular, because of the Service Oriented Architecture paradigm, it is often not viable in environments with high numbers of low bandwidth feeds.

Japan's High Energy Accelerator Research Organization (KEK) experimented with Objectivity/DB on a high latency, low (2Mbits/second) satellite network. They concluded

that improvements would be needed to Objectivity/DB to operate a useful system over the satellite link. Their research is summarized at http://tinyurl.com/34z832.

Developers could theoretically reduce the effects of unreliable networks by catching errors and retrying the operation, particularly Commits, but this places an unacceptable burden on them.

What languages must support this capability?

- At least .Net for C# and Java.
- Ideally, C++.

Which platforms must be supported?

- Tier 1 and 2 platforms.

Do any competitors already have this feature?

- RDBMSs are inherently able to operate in lower bandwidth networks.
- Versant uses an object server model, so it can operate in lower bandwidth networks.

Customers who require this feature

- Data fusion customers, particularly close to the sensors.
- Geographically dispersed applications.
- Some Internet enabled applications.

Revenue at risk, or which could be won

- We lost potential business at Green Cargo because of the requirement for high bandwidth networks.
- Having the option to operate in higher latency networks will improve the resilience of the product in grid environments.

When is this required?

- Release 11.

Additional Notes

1. We will also need:

- Additional marketing collateral
- Sales training material
- New training material
- Additional Quality Assurance material
- Additional instrumentation

2. The combination of the following techniques could improve performance in high latency, medium bandwidth, and unreliable networks:

- Compression
- Checksumming to verify message integrity
- Message sequence numbering.
- Batching AMS calls into fewer, longer messages.
- AMS look-ahead disk reading, e.g. when scanning containers.
- Automatic message retries.
- Dual channel messaging (sending the same message to two ports and comparing the incoming streams).

3. A message based API was first requested in the Objectivity/Grid Peer-to-Peer MRD (12/05/2000). However, it is a lower priority than the other two requested features (high latency, unreliable networks).

4. Deferred replication – allowing delayed replication across slow or unreliable networks and Container Replication – making it feasible to have tiered storage hierarchies (e.g. central, regional, in-store and personal) will be covered in an improved High Availability MRD.