MARKET REQUIREMENTS DOCUMENT

Feature Name: **Open Server Interfaces**

Version: 2 **Date:** 5/5/04

Completed By: Leon Guzenda

Description of the Problem

Background

Objectivity/DB uses a distributed processing approach to enhance performance, scalability and reliability. It does this by dividing the work of the DBMS engine into a local library and two kinds of remote server process – lock servers and page servers (an Advanced Multithreaded Server or an nfsd daemon process). The lock server interface protocol is proprietary. The page server interface protocol is a proprietary superset of the NFS protocol.

The AMS provides database replication functionality in addition to file and fixed size data block [page] access. Minorities of the users require the database replication functionality. However, writing to a remote file via the AMS is measurably faster than using an nfsd daemon, so it tends to be heavily used in distributed system configurations.

Several users have requested enhancements to the AMS to help them measure performance or tune it to their actual patterns of file and data usage. Some of these have been incorporated in performance releases. Others have been proposed as a fee. Another option, for advanced users only, is for them to purchase the Object Oriented File System [OOFS] option and modify the interface to the underlying file system. OOFS was provided as a part of a package of changes made to accommodate use of the industry standard High performance Storage System [HPSS]. To date, the only known adopter of this option is SLAC.

The Problem

Performance: The currently released version of AMS is still running at a fraction of the capability of advanced file system and hardware technology. SLAC produced its own version of a page server to support their homegrown adaptation of the ROOT “DBMS”. The SLAC file server is considerably faster than AMS. Recent changes to AMS may negate this difference. Nevertheless, the current AMS/OOFS combination does not appear to offer the flexibility that Very Large Database and secure sites are likely to need in the near future.

New technologies and platforms: Users cannot rapidly integrate Objectivity/DB with existing technologies, such as Lawrence Berkeley Laboratory's Distributed Parallel Storage System [DPSS] or emerging Object File Systems, such as the one from Panasas, Inc. Likewise, they cannot run an AMS process on a platform that is not officially supported by Objectivity.

Description of the Requested Feature

Functionality

We should make it possible for users to replace an AMS server with a page server of their own choice To do this we need to:

- Make the client –AMS interface protocol documentation available to licensed users.
- Provide a compliance test that will allow them to verify that they have correctly implemented a replacement server.
- Provide regular support for Objectivity products running on a compliant server.

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

This facility will be selectively licensed to advanced users.

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

The current solution is AMS plus OOFS. However, the AMS layer seems to be inefficient when compared with more specialized servers.

What languages must this capability support?

- The capability is transparent to the current APIs. However, the current client to AMS protocol is implemented in C++.

Which platforms must be supported?

- All currently supported platforms, starting with the ones that SLAC licenses.

Do any competitors already have this feature?

- No.

Customers/Prospects who require this feature

- SLAC.
- Secure sites (possibly).

Revenue at risk, or which could be won

- Ongoing revenue from the BaBar collaboration.

When is this required?

- Release 10 for general availability.
- As soon as possible for SLAC.

Additional Notes

1. We may need a special contract that will specify exactly how our IPR is to be protected.
2. We should consider replacing the current client-AMS protocol with a grid enabled protocol before implementing this capability.
3. We should consider publishing the client to lock server protocol at some later date, but only after a reimplementation of the lock server to incorporate new paradigms and to add user hooks similar to OOFS.
4. In some cases the clients can be directly attached to a SAN that looks exactly like a local file system, avoiding the need for AMS. Those users (such as Northrop Grumman and SGI/CXFS customers) will not need the proposed feature for performance reasons. However, they may still want to substitute a proprietary, secure server.