Market Requirements Document

Feature Name: Mac OS X Port

Version 1: Date Submitted: 02/2210

Completed By: Leon Guzenda

Description of the Problem

Apple Mac OS X is a Unix variant that is solely deployed on Apple Computer hardware. It has become increasingly popular in workgroup and high end computing clusters, but Objectivity/DB doesn't run on it.

Mac OS X is similar to BSD, but it uses its own Objective C compiler, which provides a unique object-oriented development environment that is similar to C++ and Smalltalk. This makes the port a little more difficult than a regular C++ or Java port.

Description of the Requested Feature

A full port of Objectivity/DB to Mac OS X, including:

- The kernel library.
- Development and database administration tools.
- An installer with the look and feel that Mac users are familiar with (see Notes).
- Synergy with built in Mac OS backup features (Time Machine).
- Full documentation.
- Web based training and samples and other information in Devnet.
- Marketing collateral.

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

• It will be a regular API, with enhancements to the installer and backup/restore.

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

Apple developers can't currently use Objectivity/DB, other than via ODBC, which can't exploit any of the power of the product.

What languages must support this capability?

- C++ (Essential)
- Java (Soon)
- SQL++ (Eventually)

Which platforms must be supported?

• Apple Mac OS X – Intel variants and (possibly) iPhone/iPad hardware.

Do any competitors already have this feature?

- The Versant db4objects C# product can be hosted by Mono running on Mac OS X, but performance may be an issue.
- Many RDBMSs have Mac OS X variants.

Customers who require this feature

- People using Mac OS X in entertainment, scientific, engineering, visualization, medical and marketing (data mining) applications.
- We have a prospect who wants to build a bioinformatics/healthcare product on Mac OS X using Objectivity/DB and is prepared to help develop/test the port.

Revenue at risk, or which could be won

- The Mac OS 9 platform was deemed important enough to merit a port to Mac OS 9 at Objectivity/DB Release 4.0 (<u>http://bit.ly/cwLhYi</u>). We also delivered, but never sold, a NeXT (similar to Mac OS X) variant.
- The MAC OS X platform now commands about 5.16% of the market desktop, i.e. 5x that of Linux, which we already support. However, it has an insignificant share of the Unix server market, which is dominated by Sun, HP and IBM. It is likely that most of the new revenue will come from the workgroup (desktop) and mobile market (if we deliver an iPhone/iPad OS variant).

When is this required?

• Release 10.1 or 10.2.

Additional Notes

1. Related Material

We will also need:

- A release note.
- New/ported Quality Assurance material.
- Hardware for the development engineer(s) and QA. Many System Engineers already have Mac OS X on their laptops or at home.

2. iPhone OS and the iPad

Although it uses the same development language and environment (Xcode) the iPhone OS has a distinctly different user interface. Although current iPhones have limited storage and power, it would be convenient to have a "lite" runtime version of the Mac OS X product for the iPhone, or a client front end. Although the market share isn't yet predictable, it is likely that the iPad, which uses iPhone OS, will have more than enough power to run full-fledged applications based on Objectivity/DB.

3. Alternative Porting Strategies

As noted above, db4objects uses the Mono environment to allow C# applications to run on Mac OS X. While this is less likely to perform as well as a direct port, it may be competitive with a C++ to Objective C layer and be faster to implement.

4. Installers

"Installer is an application included in Mac OS X (and in its progenitors <u>OPENSTEP</u> and <u>NEXTSTEP</u>) which extracts and installs files out of .pkg packages. It was created by <u>NeXT</u>, and is now maintained by Apple Inc. Its purpose is to help software developers create uniform software installers.

Installer resides at /Applications/Utilities/Installer.app on Macintosh computers running MAC OS X versions prior to 10.5, and launches when a package or metapackage file is opened. In Mac OS X v10.5 the Installer application is located at /System/Library/ CoreServices/Installer.app.

The installation process itself can vary substantially, as Installer allows developers to customize the information the user is presented with. For example, it can be made to display a custom welcome message, software license and readme. Installer also handles authentication, checks that packages are valid before installing them, and allows developers to run custom scripts several points during the installation process.

There is also a command line version located at /usr/sbin/installer." - Wikipedia.

The PackageMaker tool is available within Xcode. The documentation is easy to follow and a simple installation can be created very quickly.

[Continues...]

5. Backups

Mac OS X has a very simple backup restore system and interface named Time Machine, based on a versioned file system. An individual machine can be connected to a local or remote disk which can be selected as Time Machine's backup disk. The application runs in the background, periodically saving updates to every file that gets changed.

Simple as it is, there could be problems if this mechanism is used in conjunction with Objectivity/DB, particularly when databases are frequently or continuously updated. This is because an application can limit the number of file descriptors used by the operating system at any time. This feature was introduced to overcome limitations of early Unix operating systems, which would often be configured to allow no more than 16 or 32 open files per process.

The Objectivity/DB Storage Manager can close and reopen a file during a transaction. Although the changes are not committed, in an ACID transactional sense, an updated file has been closed, as far as the Operating System in concerned, so it may be a candidate for a backup. Restoring that version of the (possibly partially updated) file later could destroy the integrity of a container, database or federation.

It may be possible to turn off Time Machine for Objectivity/DB volumes, or to have a background process that opens a file for update at the request of a process/thread that needs to close it, then closes it after the process has reopened it. That may prevent Time Machine from backing the partially updated file up, but the topic needs thorough investigation. This isn't only a problem on Mac OS X, by the way. It can occur with any background backup mechanism.