Market Requirements Document

Feature Name: Objectivity/Lite

Version: 5 **Completed By:** Leon Guzenda **Date Submitted:** 03/16/06

Description of the Problem

Objectivity/DB development and runtime licenses are quite expensive, especially for simple, high volume deployments. The current Objectivity/DB licensing arrangements lead to a situation where application developers looking for a persistent object repository are more likely to select low end or open source products, such as FastObjects or db4objects. We could simply introduce new low end pricing, but existing users could challenge this strategy as they could claim that they are being unfairly penalized, especially as low price licenses would be using exactly the same product as the high price licenses.

Description of the Requested Feature

We need an entry-level product that will encourage new users to start out a product that has a clear and simple upgrade path. However, it must not be too functionally capable, in order to protect our mainstream licensing model.

There should be a subset of the existing API that would support essential features and exclude use of the advanced features. It should be upwardly compatible, merely requiring re-linking with the corresponding full product. It could have more than one variant, e.g. a subset API with additional limitations, such as on the amount of data that can be stored. It is intended that Objectivity/Lite be primarily purchased via our online site.

At least the following features of the regular products may be omitted:

- Versioning
- High Availability (Autonomous Partitions and data replication)
- Active Schema.
- Propagation of delete() and lock() operators.
- SQL++ syntax within iterators
- Legacy (deprecated) interfaces.
- Object naming (To Be Determined)
- Scalable collections.
- Automatic recovery by the lock server.
- Iterators that go more than one level lower in the storage hierarchy (e.g. iterating over all of the objects in a database).
- DBA interfaces

Additional APIs and options are acceptable. Note that reducing the number of features in the API may reduce the footprint, making it easier to do a real-time port later, but this is not a primary goal of Phase 1 of Objectivity/Lite.

We want to reduce the possibility that users could build extensive applications without buying appropriate licenses. So, it must be possible to limit:

- The number of databases that a federation can hold (maximum of 2, plus the system database).
- The number of concurrent transactions (10).
- The total number of schema classes and subclasses (20).

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

This feature is primarily a subset of an existing feature, but it may require extra build, quality assurance and documentation effort. If we add ways of limiting usage, e.g. by preventing use of more than one database, then it will require extra kernel and QA effort.

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

Partners must learn to use the full product. We are unwilling to deeply discount our standard product. New users are increasingly flocking to open source object databases, such as db4objects.

What languages must support this capability?

- Java first, then C# or C++ later.

Which platforms must be supported?

- Tier 1 platforms, starting with Windows XP and Linux, then others as required.

Do any competitors already have this feature?

- ObjectStore (PSE)
- Oracle (including a new grid enabled version and the recent acquisition of BerkeleyDB)
- IBM (a free, embeddable version of DB2 and the open source Cloudscape)
- Sybase
- Versant (FastObjects, formerly known as Poet)
- db4objects (see Appendix A)

- ObjectDB
- MySQL and PostgreSQL (open source) could be regarded as competitors, but they don't usually compete in our markets.

Customers who require this feature

- Potential object oriented software partners.
- Startups
- Developers of simple object oriented applications that are finding it hard to map their object model to RDBMSs.
- Educational establishments.
- If we can reduce the footprint, telecom, process control and defense VARs who want to move the database out into the field (nearer the "edge").

Revenue at risk, or which could be won

- Over the years we have missed opportunities at Rational, Teknowledge and Informix because of the effort and cost involved in porting products to Objectivity.
- Having more partners, especially in the Federal and bioinformatics markets, could significantly increase the rate of adoption of our products.
- Some proportion of the revenue that is flowing to FastObjects, db4objects and ObjectStore PSE could be ours.
- We should look at this product as a zero profit, market broadening tool. However, it might attract a large audience that we've previously been unable to tap.

When is this required?

- Release 9.3, or earlier. (end-July 2006?)

Additional Notes

1. We will also need:

- Marketing collateral, including promotional material and a special area on our web site.

- Technical Publications.

- New QA material to prove that the reduced API works and that all other standard API functionality does not work.

- The Functional Specification should include a copy of the current API with all features that are to be omitted struck or shaded out.

2. It may be possible to use one bit in the license key to signify that the product being used is limited. The subset product could then have two variants – "inhibited (or governed) subset" and "subset".

3. Licensing costs are to be determined. They should be realistic and should attempt to cover actual development and support costs.

APPENDIX A – Licensing and Support for db4objects

db4objects unrestricted runtime licenses have a mean unit price of $9 per runtime to as low as $1. They have a developer support package for $1,200 per member per year, described as follows:

"The **db4o Developer Network (dDN)** membership is our solution for companies that have decided to develop software with db4o or run a db4o based application in house. The dDN membership package includes:

- Personalized, private and tracked 24/7 case support through our exclusive dDN Member Portal (screenshot) with a response time of 24 hours or less
- A lock-in of any negotiated rates and conditions for all future purchase of dRT db4o runtime licenses, thus allowing fixed budget assumptions in your product planning
- Free updates on any previously purchased or distributed volumes of dRT db4o runtime licenses
- An unlimited number of non-GPL developer licenses within your organization"