

## Market Requirements Document

**Feature Name:** Objectivity for J2ME

**Version:** 2    **Completed By:** Leon Guzenda    **Date Submitted:** 09/26/06

### Description of the Problem

Objectivity for Java is implemented on Java 2 Platform Standard Edition [J2SE]. The Java 2 Platform Enterprise Edition [J2EE] adds extra APIs for server-side applications. Objectivity for Java includes support for J2EE Connector Architecture [JCA] mechanisms. However, mobile devices tend to use Java 2 Platform Micro Edition [J2ME], which implements a subset of the API provided by J2SE, so it may be impossible to run our current Java product on it.

### Description of the Requested Feature

Objectivity for J2ME should be a fully functional variant of Objectivity that runs on the Connected Device Configuration [CDC] variant of the J2ME platform. A reduced API (covered in Appendix A) would be acceptable in order to reduce the runtime footprint of the product.

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

This feature is a rework of the existing Objectivity for Java. It is essentially a port, but it may require the removal of some standard features and the addition of others.

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

Users of J2ME, which will include telecom, process control and instrumentation equipment vendors, cannot currently use Objectivity for Java directly. However, they may be able to use Java Database Connector [JDBC] for a relational view of a remote Objectivity for Java federation.

### What languages must support this capability?

- Objectivity for Java.

### Which platforms must be supported?

- Solaris and Windows (XP or Vista.)

**Do any competitors already have this feature?**

- DB2 Everywhere [DB2e]
- McObjects Perst Lite
- Mdrawer (Open Source).
- Oracle10g JDeveloper - J2ME Extension
- PointBase Micro.

**Customers who require this feature**

- ILS Technology.
- Developers of mobile object oriented applications that are finding it hard to map their object model to RDBMSs.

**Revenue at risk, or which could be won**

- We are not aware of any business lost as a result of not supporting J2ME.
- A J2ME port would make it possible for equipment vendors and defense application builders to move Objectivity applications further out into the field.

**When is this required?**

- Release 10.1.

**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 API works and is interoperable with other platform and language combinations.

2. Licensing costs are to be determined.



## **APPENDIX A – Reducing the footprint of Objectivity/J2ME**

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.