Market Requirements Document

Feature Name: License Manager

Version: 1 **Date:** 3/19/02

Completed By: Leon Guzenda

Description of the Problem

The current product does not enforce the License Agreement. This means that Objectivity's Intellectual Property Rights may be exploited by: an evaluator; a person who downloads a trial version; or a customer whose usage patterns change. It is also possible for a user to deploy unlicensed components, such as FTO or DRO, without paying for them.

We recognize the fact that our customers do not wish this to be an intrusive facility. Likewise, any software based licensing scheme is open to circumvention. For instance:

- Developers may perform a remote login to a licensed machine, run development tools there and then move the results back to their own machine, or
- They may run multiple Federated Databases to reduce the apparent concurrency; or
- End users may patch the kernel library object code to make it skip around the license checks.

The mechanism should be robust but it does not need to be bulletproof. At worst, we will be in our current situation.

Description of the Requested Feature

The kernel should issue warnings and then issue fatal error messages at every API call or tool invocation once a predetermined set of conditions is met. These conditions would not be modifiable by a user other than by applying a new 128-bit (or greater) key issued by Objectivity. The conditions must include at least the following options:

1. A maximum number of concurrent processes or threads accessing a single Federated Database.
2. An expiration date.
3. A license granted to a single hostname or username. [This would be the default for developer tools].

There would be at least four new tools:

1. A license key generation tool that can only run within Objectivity's domain. The keys would contain option and parameter sets generated for a particular license.
2. A tool for installing or upgrading a license at a site.
3. A tool for monitoring license usage at a site.
4. A tool that would allow VARs to issue sublicenses to their customers from a specific domain. They would also supply tools 2 and 3 to their customers.

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

- All of the features would be included in the standard and downloadable products.

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

We rely on the good faith and best endeavors of our customers. We have at least one licensing dispute per Quarter. We have the right to conduct audits, but it would be easy to modify usage or environments before an audit occurred.

What languages must support this capability?

- Mandatory in first release: C++ and Java
- Optional in first release: Smalltalk and SQL++.

Which platforms must be supported?

- All current platforms, but we should focus on the Tier 1 platforms first.
- The tools must be platform independent. [Note, this could influence any encryption algorithms that are used within the scheme].

Do any competitors already have this feature?

- Oracle.

Customers who require this feature

- We require it more than our customers.
- Some VARs will probably require this feature.
- It may be useful to give VARs hooks for adding their own conditions. These hooks would be subservient to our overall policies.

Revenue at risk, or which could be won

- License disputes rarely yield less than tens of thousands of Dollars. We don't know how much revenue we are losing.

When is this required?

- Release 8.

Additional Notes

We will also need:

- Amendments to our License Agreements.
- Sales and DBA training material
- Updates to the appropriate technical publications.
- A program for ensuring that current customers migrate to the release that introduces the License Manager.
- Additional mechanisms within Administration and Customer Support.

Notes for the designer(s):

- **The mechanism must be deployable at secure sites.**
- If feasible, the license information should be encrypted and held at the Federated Database scope.
- The kernel and the servers may both be made responsible for checking the license, but it would be acceptable to have only the servers check the license if they can fully (possibly jointly) apply the checks set forth in the License Agreement.
- It might be feasible to deploy a Super Federation as a standard configuration, with one Federation being solely used for license enforcement and performance statistics archival.