Market Requirements Document

Feature Name: Event Notification

Version: 1 **Completed By:** Leon Guzenda

Date Submitted: 06/18/07

Description of the Problem

Background

Applications and users frequently want to be notified when objects or other structures in a federation are created, updated or deleted. The only ways that this can happen currently are:

- Objectivity/SQL++ triggers, but the application writer has to add their own mechanism for signaling another user or process.
- An application that has cached an out of data page will receive an exception if it attempts to obtain an update lock on the page's container.
- A sample class library written by a Systems Engineer implements an event notification service based on the Objectivity for Java API.

Problem

All of the above mechanisms essentially require significant modifications to application code or inefficient polling of a structure used to monitor changes.

Description of the Requested Feature

1. **Scoped Topics**: A mechanism for a Publisher to define Scoped Topics, i.e. transient or persistent named logical channels associated with one or more object classes and a scope (individual, container, database or federation). Examples might be:
    a. "Newly_Created_Alarms". // Associated only all Alarm objects.
    b. "Deleted_Containers" // Associated with all containers in a database or federation.
    c. "Deleted_Container_We_Used_Yesterday // Associated with a specific container.
    d. Updated_Incident // Associated with Incidents in a particular database.
2. **Content Filtering**: A mechanism for a Subscriber to register interest in a Scoped Topic and to optionally set filters for including or excluding items based on their contents or other properties (e.g. number of objects in a container).

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

1. Event Notification will be a new feature.
2. It may be able to leverage some of the kernel event listeners that have been built for debugging and instrumentation purposes.
3. It should be able to leverage a Distributed Queue Manager (a separate MRD).

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

Many applications need to poll infrequently updated items or to immediately notify a user or process when a new object arrives, or a condition is met. Polling is inefficient, particularly for very large databases.

What languages must support this capability?

- All APIs, starting with C++, Java and .Net for C#.

Which platforms must be supported?

- At least the Tier 1 platforms.

Do any competitors already have this feature?

- GemFire (GemStone)
- JADE
- ObjectStore.
- Polyhedra
- Versant

Customers who require this feature

Many customers, particularly in the “Monitoring, Analysis and response” category have built their own event notification mechanisms or used middleware. Typical users would include:

- Engineering/Manufacturing
  - Computer Aided Design/Engineering/Manufacturing (CAD/CAE/CAM)
  - Configuration Management
  - Data Acquisition & Telemetry
  - Process Control & Automation

- Complex Financial
- Geographic Information Systems [GIS]

- Government
  - Control, Communications, Computers. Intelligence, Surveillance, Target Acquisition and Reconnaissance (C4ISTAR)
  - Data Fusion
  - Intelligence Community
- Health Sciences and Medical Equipment
- Scientific Computing
- Internet Related
  - eCommerce
  - Internet Infrastructure
  - Software as a Service
- Complex IT, especially Complex Event Stream Processing
- Metadata Management
  - Content Management
  - Knowledge Management
- Telecom
  - Configuration Control
  - Advanced Intelligent Network
  - Element Management System| Network Element Management System
  - Network Management
  - Network Operations Center
  - Operations Support Systems
  - Personal Communications Service

Revenue at risk, or which could be won

- An efficient event notification mechanism should increase our chances of winning business in the following market categories:
  - **Design and Simulation**
  - **Metadata and Knowledge Management**
  - **Monitoring, Analysis and Response Systems**

When is this required?

- Post Release 10.

Additional Notes

1. The event notification capabilities could leverage a Distributed Queue Manager if it is implemented in time.

2. It is important that the event notification mechanism not impose an undue performance burden on critical operations. This may mean invoking asynchronous functionality running in threads other than the client thread that causes an event.

3. 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.

4. Licensing costs are to be determined.