Market Requirements Document

Feature Name: **Instrumentation**

Version: 4 **Date:** 07/19/2007 – Only the Notes section has changed since Version 3.

History: Version 1 - 01/26/2005
Version 2 - 05/23/2007
Version 3 – 05/23/2007

Submitted By: Leon Guzenda

Description of the Problem

Some, but not all, Objectivity features are instrumented in order to allow users to monitor the status, behavior and performance of a process, server or feature. This makes it harder to find performance bottlenecks and to monitor the status of a live system.

Description of the Requested Feature

This enhancement will provide instrumentation tools and interfaces for monitoring the runtime behavior of Objectivity applications with minimal impact on system performance. At least the following mechanisms should be added:

- A page server monitoring tool that would provide, at minimum:
  a. The frequency at which physical pages are read and written sequentially.
  b. Rankings showing the frequency of access to databases, containers and the most frequently accessed logical pages.
- Listener interfaces, similar to the existing one for the Lock Server, that allow an external process to monitor activity.
- Persistent client and server side statistics that could be queried using Objectivity/DB APIs and Objectivity/SQL++ (including ODBC).
- Lock Server metrics: A change to the Lock Server Monitor API to permit limiting the activity reporting to specifically designated databases or containers.
- A transaction correlation tool that shows the sequence and timing of transactions issued by specific or all threads/processes and similar information for specified or all containers/databases.
- Java Management Extensions (JMX) to monitor lock server, AMS and Query Agent activity. The JMX Agent would encapsulate the listener interfaces for the lock server, AMS and Query Agents.
- Graphical plug-ins to Assist to aid the user in visualizing the above metrics.

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

- Enhances the kernel and adds new tools and plug-ins.

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

Application developers have to add their own performance monitoring facilities, or request assistance from Objectivity System Engineers or Customer Support staff. Difficult problems often require the attention of Engineering developers.

The lack of adequate instrumentation decreases customer satisfaction, increases our overheads and adversely affects Engineering schedules.

What languages must support this capability?

All APIs, as the capabilities are all to be provided by the kernel. C++ and Java should take precedence over Python, Smalltalk and SQL++.

Which platforms must be supported?

All platforms.

Do any competitors already have this feature?

Oracle, DB2, Sybase, ObjectStore and Versant.

Customers who require this feature

All.

Revenue at risk, or which could be won

Performance tuning accounts for a significant proportion of many development efforts and is a prime source of customer dissatisfaction with deployed products. Evaluators will be more likely to choose a product that has instrumentation.

When is this required?

- Release 10.

Additional Notes

1. We will also need:

- Marketing collateral
- Sales training material
- New training material
- Additional Quality Assurance material

2. Dashboard widgets from Sherrill Lubinski Corporation (SL) could be used for rapid implementation of the Assist plugins.

3. We should also review the existing statistics for completeness. There are probably also other statistics that should be added. For example, two that came to light in recent customer issues are that:
a) We don't count object migrations.
b) We don't account for memory allocation for page maps
or copy space.

4. The Release 10 "listener" interfaces for the AMS, PQE and clients can be very simple. The servers could support extra calls to reset statistics and get statistics. The client could simply write statistics to stdout (or a new file) to be read by the management agent.

5. The statistics required from the AMS are:

- Total number of pages read since the statistics were reset.
- Total number of pages written since the statistics were reset, excluding resync activity (if possible).
- Total number of replicated pages written by resync.
- For each file read or written since the last reset:
  - Number of pages read.
  - Number of pages written.

6. The statistics required from PQE are:

- Total number of query requests received since last resync.
- Total number of objects scanned since last resync.
- Total number of objects passed to filters (plugin) since last resync.
- Total number of qualified objects returned to clients.
- Object types requested, together with the number of requests, number of objects scanned and number of qualified objects returned per type.