Market Requirements Document

Feature Name: Extra Debug Facilities - Snapshot

Version: 2 **Date:** 01/31/2005
Version: 1 Date: 01/26/2005

Submitted By: Leon Guzenda

Description of the Problem

The Release 9 client side API monitoring interface is a considerable improvement over its predecessor (trace mode). However, Objectivity/DB runtime errors are very difficult to diagnose after the fact, especially in heavily loaded, concurrent applications. Specifically:

- Many of the current diagnostic mechanisms either show a historical statistical summary [ooRunstatus] or a static situation [oospace, oolockmon].
- Error messages are useful to Objectivity engineers but don't provide many clues to the application developer as to their probable cause.
- Diagnostic tools are limited to a single thread/process or server. It is difficult to interpret interactions between clients.

Description of the Requested Feature

This document requests a new feature that will enable any client thread/process or an administrator to request a snapshot of vital Objectivity/DB state information.

If an exception occurs in a client application there is very little information available about the state of the client thread/process and the associated servers. The kernel should be enhanced to optionally provide:

- A snapshot of the current date/time and Objectivity software version.
- A snapshot of the currently set transaction/session variables.
- A snapshot of the currently open databases, containers and objects, indicating whether they are open for read or update.
- A snapshot of the state of the cache providing enough detail to know: which cache pages are currently in use; whether or not they are dirty; whether or not they have been previously written during the current transaction; the logical #DB-OC-PN in each page; and the contents of the two most recently accessed pages.
- A snapshot of all statistics maintained for ooRunStatus().
- A snapshot of all active file descriptors, correlated with the databaseIDs.
- A snapshot of all active Objectivity/DB TCP/IP connections and their purpose (e.g. connection to Lock Server on HostX, connection to AMS on HostY).
- A snapshot of any internal trace information (such as the circular buffer for tracking API calls) that may be useful to Kernel engineers.

- Each of the above snapshots may be enabled via an environment variable. They may also be enabled or disabled via API calls.
- The destination for the snapshots will be a file defined via an environment variable or an API call.
- The application may request a snapshot at any time.
- There should be an optional plug-in for the ooAssist framework that will take a snapshot of a thread, a process or a process and all of its threads and display the snapshot contents in a user-friendly manner.
- An API that enables a thread or process to freeze all current activity for that process and snapshot the state of the process and all of its threads.
- An API that enables a thread, process or DBA tool to freeze the activity of all associated threads and servers. It then snapshots the state of the process and all of its threads and causes the associated servers to snapshot their state.
- An API and a DBA tool that suspends the action of a particular lock server and causes it to snapshot the information available via oolockmon and oolockwait to a designated file.
- An API and a DBA tool that releases the suspension of a designated lock server.
- An API and a DBA tool that suspends the action of a particular AMS and causes it to snapshot information about its current connections and file descriptors to a designated file.
- An API and a DBA tool that releases the suspension of a designated AMS.

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

- Snapshots (client and system wide) - New APIs, runtime options and tools.

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

Application developers have to add their own diagnostic facilities, or request assistance from Objectivity System Engineers or Customer Support staff. Difficult problems often require the attention of Engineering developers. Some problems have been impossible to reproduce within Objectivity’s engineering environment.

The lack of adequate diagnosis capabilities 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.

Which platforms must be supported?

All platforms.

Do any competitors already have this feature?

Oracle, DB2, Sybase and ObjectStore (limited).

Customers who require this feature

All, particularly customers who have appeared on the weekly Warm and Hot lists.

Revenue at risk, or which could be won

Diagnosing errors accounts for a large proportion of most development efforts and is a prime source of customer dissatisfaction with deployed products. Every serious problem in a deployed application increases the risk of losing a customer. Evaluators will be more likely to choose a product that has good development and deployment tools.

When is this required?

- Release 10.

Additional Notes

We will also need:

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