Market Requirements Document

Feature Name: Objectivity/Administration Console

Version: 0.1	Date Submitted: 3/19/15	Completed By: Brian Clark
Version: 0.2	Date Submitted: 3/19/15	Completed By: Brian Clark
Version: 0.3	Date Submitted: 7/21/2015 Added expectations in additional net	Completed By: Brian Clark otes.

Description of the Problem

Objectivity/DB, when installed and made operational, consists of a number of components, likely distributed over the network, that work together to provide the ODBMS functionality. However, there is no single point that lets users visualize and manage their entire Objectivity/DB enterprise. This MRD addresses the need for this capability.

Terminology

Wikipedia defines a dashboard as "an easy to read, often single page, real-time user interface, showing a graphical representation of the current status (snapshot) and historical trends of an organization's key performance indicators to enable instantaneous and informed decisions to be made at a glance". This is limiting and only part of what we are trying to do.

An administration console however provides system administrators and advanced users an interface for configuring and monitoring the system.

I prefer the use of console.

Background

There have been several attempts at providing a single GUI to access Objectivity/DB information. These have used different frameworks, but have only addressed part of the needs; there has never been a complete solution.

There are many tasks that are common to installing, deploying and managing an Objectivity/DB environment. They environment itself may include a single operational federation running on a single machine, or it may consist of numerous federations that must be managed. The console should be designed to support both small scale single

federated database deployments and large scale, distributed multi-federation deployments.

Description of the Requested Feature

A single point of administration that lets users visualize and manage their entire Objectivity enterprise deployment.

Intended users are database designers, developers and administrators; schema designers, developers, and administrators; system planners and operators.

An Objectivity installation is comprised of processes (client applications, services) and files (databases and configuration) in a distributed network environment. Any computer in the network running any Objectivity process or hosting any Objectivity file, including installed Objectivity software, will be considered an Objectivity host. Different views into the installation across multiple hosts will be required.

Network view

The Network View is focused on the distributed compute environment. The network view will present the user with a list or graphic, or both, showing all of the hosts that have been configured to participate as Objectivity hosts. From this view users will be able to monitor every host's availability on the network as well as the status of Objectivity services expected to be running on each host. The use will no longer have to run the various "check" commands for each host.

Host View

The Host is a single host view. It presents to the user information relevant to the operation of Objectivity software, processes and storage of Objectivity data on a single host computer.

Managing Services: The host view allows the user to view status, deploy, start, stop, and configure services on the currently selected host.

Instantiating new federations: The host view allows the user to create a new Objectivity federation.

Viewing and Provisioning storage: The host view allows the user to view the storage volumes (disks) on the currently selected host. The use will be able to see what Objectivity Storage Locations are currently defined on the host, the sizes of the database files and the amount of available disk space.

The user will NOT be able to define new Objectivity Storage Locations or Zones from this view as that is a Federation view capability.

Federation View

The federation view allows the user to manage all aspects of a federation's configuration.

Database File Management: The Federation view allows the user to move database files from one location to another.

Service Management: The federation view allows the user to perform federation centric service management such as moving the lock server from one host to another.

Tool configuration and execution: The federation view allows the user to configure and execute Objectivity tools.

Schema View

The Schema view allows the user to perform all schema related tasks.

View schema:

Edit/evolve schema:

Import/export schema:

Placement View

The Placement view allows the user to view, create, edit, and monitor all placement related configuration details for a federation.

Storage Location and Zone monitor:

Storage Location and Zone editor:

Placement Rules Monitor:

Placement Rules Editor:

Index Management:

Simulation:

Replication View

The Replication view allows the user to define and deploy replication strategies within the context of a managed federation.

Replication strategy manager:

Replication strategy monitor:

Replication strategy editor:

Task Monitoring & Control View

The Task Monitoring & Control view allows the user to monitor and control specific tasks performed by the Objectivity Tasking framework e.g. InfiniteGraph pipelining.

Backup View

The Backup view allows the user to create, deploy, and execute backup and restore strategies on a managed federation.

Backup strategy monitor:

Backup strategy editor:

Restore tool:

Query View The Query view allows the user to develop and execute "DO" queries.

Application Configuration View

The Application Configuration view allows the user to configure application configuration parameters, both Objectivity parameters and application specific parameters (properties).

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

This is a new feature, gradually replacing existing GUI components.

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

A variety of GUI tools attempt to provide subsets of this functionality.

What languages must support this capability?

Implementation of this feature is independent of any language API.

Which platforms must be supported?

All.

Do any competitors already have this feature?

Yes, many traditional databases, NOSQL and Big Data solutions have some form of management console.

Customers who require this feature

All users will find it easier to get started and manage deployed Objectivity/DB installations.

Revenue at risk, or which could be won

Could lead to more early adopters, and successfully deployed installations

When is this required?

Release 12.

Additional Notes

1. Implementation notes:

- a. Platform: At a previous strategy meeting there was consensus for a web browser based approach. This means that it would not be standalone but have some kind of client/server architecture. The platform would use HTTP protocol between client and server, possibly REST, and it would not expose database sessions to the user. Platform requirements include: ease/ efficiency of use; consistency of look and feel; extensibility (internal and external); familiarity (cross OS/browser).
- b. Some sort of agent running on each host.
- c. Security across the wire may be a concern? Port security, encryption, SSL, access control?
- d. Dan Hall's document "A Views-Based Approach To The Dashboard"
- e. Strategy meeting notes.
- f. "JPM_DistributedJMXMonitoringRequirements" document.
- g. Additional views: Security; Development Tools, Visualizer integration; Assist integration.

2. Related Material

We will also need:

Marketing collateral, including a Press Release. Technical Publications. Release Note. Extra Quality Assurance Material.

3. Software requirements

a. Depends on choice of client/server implementation and agent technology.

4. Hardware Requirements

No special requirements.

5. Expectations

- a. All Objectivity/DB tools should be accessible via the web user interface.
- b. There may be multiple ways of invoking a tool; e.g.
 - i. Single click of a button will invoke the tool with all default values;
 - ii. A way to display a form with default values, and a way to edit these values before submission.
- c. Some ways to display the tool help text.
- d. Use the same names for display parameters as the tool uses, e.g. "CreateFd

- -fdName fdName [-lockServerHost hostName] [-fdDirHost fdDirHost] [-fdDirPath fdDirPath] [-jnlDirHost hostName] [-jnlDirPath jnlDirPath] [-licenseFile licenseFile] [-noPlacement] [-noTitle] [-help] [-quiet]"
- e. Use the same names for any output display as the tool uses.