

Market Requirements Document

Feature Name: Internet Friendly Features (Objectivity/i)

Version: 2

Date: 12/22/99

Completed By: Leon Guzenda

Description of the Problem

At least four of our competitors (ODI, POET, Versant and Cloudscape) have targeted the Internet market. They have added features and are deploying resources to expand their presence in that market. We have minimal VAR penetration (Intraspect and a few other customers) in the Internet market. To date, no significant web site, ISP or Internet equipment vendor, with the possible exception of Nortel, has chosen Objectivity/DB. This is partly a question of visibility or market confusion and partly one of product features and interfaces.

This document is intended as a vehicle for exploring the full implications of trying to enter this market. I am not suggesting that we immediately divert attention from our core competencies or markets. However, it might be one component of our pre-IPO planning and post-IPO activities.

Description of the Requested Feature

I have split the requested features into five groups:

- Web look and feel
- WAN/Internet support
- Options (frameworks, class libraries etc.)
- Performance
- Marketing

Web Look and Feel

This is essential in the Internet market. New standards, protocols and conventions emerge frequently and leading edge products incorporate them very rapidly. To date, the only Objectivity features that fall into this category are the Java API, HTML documents and some degree of JDBC support. We should add:

- a) A browser version of all tools.
- b) Browsable tools integrated with a neat configuration/statistics visualization tool.
- c) EJBs for all tools
- d) HTML versions of all documents
- e) Downloadable product components - small & smart (automatically upgradeable).
- f) ODMG compliant Java APIs (64-bit integers/times and Unicode support)
- g) An ultralite client API that interfaces to a Database Server (basically, separating the current Language Interface and the Kernel).
- h) URL (DNS/LDAP) name server as a "bootfile" option.
- i) Authenticated and encrypted links between clients and servers (AMS, Lock Servers, SQL and Database Servers)

WAN/Internet support

Objectivity/DB is implemented to run in a reliable high-speed network. It has some WAN support features, such as the Fault Tolerant Option, but it performs badly or becomes difficult to administer in slow or unreliable networks, such as the Internet. These features should improve performance and usability over the Internet:

- a) Autonomous tools – making it possible to change schemas or catalogs without having to access all partitions.
- b) Partial backup/restore – removing the need to have to restore a complete federation.
- c) Deferred replication – allowing delayed replication across slow or unreliable networks
- d) WAN version of AMS – reducing the latency and (optionally) increasing the packet size of client-AMS interactions
- e) Container Replication – making it feasible to have tiered storage hierarchies (e.g. central, regional, in-store and personal)

Options

Many of the new features identified above will benefit a broad class of developers. The following optional features are most likely to be used by developers of Internet applications.

- a) XML support – the ability to create and manipulate XML; to add XML tags to existing objects and to use XML filters to access existing objects.
- b) High performance streaming interfaces (we could make Alan Ezust's Java Media Framework a fully supported product). At least MPEG, AVI, MP3 and RA should be considered.
- c) Import/export file – to allow users to capture or re-create external files safely within transactions
- d) Text Framework – a fast free-form text search API.
- e) Image Framework (GIF, JPEG etc. support, with compression)
- f) X/A Support – making it easier to co-ordinate updates to Objectivity/DB and other DBMSs
- g) ODMG Object Transfer Format (OTF) – making it easier to transport objects between federations.
- h) Transportable types – making it easier to add metadata to existing federations – may be needed to make OTF work
- i) Support for database and container level security¹.
- j) Rollforward journaling, for eCommerce sites. This would allow the application to add a transaction type/tag to a commit call, allowing controlled selective rollforward.

Performance Improvements

The improvements to the client-AMS interface detailed above are primarily needed to improve performance. However, since using our servers across the Internet is impracticable right now those features are included in the general WAN/Internet category. The following features would improve performance for systems that need to load a lot of data coming from other sources and for streamed data:

- a) High-speed data loader – this might be a tuned version of the OTF import tool. Otherwise, it would rapidly load tabular data from conventional files into one or more containers. It might also have a streaming interface, e.g. for video/audio capture.

¹ The oofs/GSA facility could suffice here.

- b) Monotyped pages – this would be a property of a container. All pages in the container would hold objects of a single type. A further refinement could constrain it to fixed length objects². Another variation could constrain the object types allowed within a container type.
- c) Streamed and direct access LVArrays – allowing the user to designate a number of pages to be dedicated to serially accessing the contents of an LVArray. It would provide better BLOB support than the RDBMSs.
- d) Fully functioning indexing on all supported data types.

Marketing

Some of the following “features” affect the marketability of the Internet friendly version of the product. Others may require changes to the packaging of the product or to our Engineering process.

- a) Zero Defect Product Philosophy
- b) Zero Learning Time – How do I use Objectivity? No problem, it’s obvious!
- c) Open source DBA tools & “Internet/multimedia” classes
- d) Downloadable product components – from free trial through post-deployment.
- e) Improved diagnostic tools and smarter Customer Support tools.
- f) Beat ODI, POET, Versant and Cloudscape in checklist features
- g) Aggressive marketing to and via Internet partners/channels and to the Java community
- h) Introduce new pricing “Superdeveloper + Free runtimes”; or “NoFeeDeveloper + monitored concurrent user license; or low/no fee developer and per-hit server licensing.
- i) Improved customer information gathering and mining tools.
- j) Optional “Guaranteed Response Time” product support, especially for bug fixes on deployed releases.

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

- Enhances Objectivity/DB, Objectivity for Java, Objectivity/FTO and Objectivity/DRO.
- Adds Objectivity/i

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

It isn't. We have an excellent Java API and some HTML documentation, but that's all.

What languages must support this capability?

- Java (essential)
- C++

Which platforms must be supported?

- At least Linux (because so many people are using Apache on it), NT, Solaris, HP, IBM and SGI.

² . The Storage Manager would segregate the user data from associations, scope names etc. The latter would be held in a separate region of the OCluster. This would completely remove most of the slot housekeeping overheads within a page.

Do any competitors already have this feature?

- ODI have most of the features that are not specific to Objectivity.
- Versant has XML and EJB support.
- Cloudscape has ultralite clients, but it is an ORDBMS.
- Oracle 8i has XML, EJB and JDBC.
- POET has XML and an object server.

Customers who require this feature

- Any VAR, ISP, portal provider or end-user building an Internet capable application that stores large amounts of complex data and/or relationships.

Revenue at risk, or which could be won

- It is likely that the major portion of ODI's revenue is now coming from its business in the Internet sector.
- All of Cloudscape's revenue is coming from this sector.
- Versant claim that their EJB initiative has paid off (maybe it has, their stock is now on a par with ODI's).
- POET are gaining strength at the low end of this market

When is this required?

- Web look and feel – Late 2000
- WAN/Internet support – Late 2000
- Options (frameworks, class libraries etc.) – XML support and streaming Late 2000. Others, mid-2001
- Performance – Late 2000
- Marketing – Zero Defects – ASAP. Others, Mid/late 2000.

Additional Notes

We will also need:

- Marketing collateral
- Qualification questions/notes for ISPs, VARs and corporate users.
- Sales training material
- A list of publications and industry events to be approached/attended
- An Internet specific seminar