Market Requirements Document

Feature Name: Export/Import Tools

Version: 1 **Date:** 3/1/00

Completed By: Leon Guzenda

Description of the Problem

It is practically impossible to selectively move large quantities of data from one federation to another dissimilar federation. The oodump and ooload tools produce bloated ASCII files and have limited functionality. The oocopydb and ooattachdb combination only works for federations with identical schemas. There is no practical way to ensure that two federations have identical schemas over an extended period. There is also no way of changing the page size of a federation.

Some of our users, such as SLAC, need multiple federations because they need a larger number of files than Objectivity/DB can currently support. Other users, such as TRW, cannot use an alternative solution, such as being able to open more than one federation in a transaction, for security reasons. Many VARs need to maintain control of some portion of their customer's federations and want to selectively ship updates to the objects that they control.

One of the proposed tools could also provide a standard way of importing large quantities of data, either for database application testing or for moving data from legacy systems. The tools would use checkpointing to minimize the impact of disruptions.

Description of the Requested Feature

This feature would add two tools to the standard database product:

- ooexport
- ooimport

The **ooexport** tool needs to:

1. Allow the user to select a group of objects to be exported to one or more output mechanisms. At minimum, the user could specify a list of databases, containers and/or collections. At best, the user could execute an SQL SELECT clause to select the objects/containers/databases to be exported. The user could further qualify the run by specifying that all associated objects, containers, databases, indices, ooMaps, collections, names and schema information be exported too.

2. Allow the user to specify an "export set" using one or more output mechanisms, including:
   - A single file or a group of automatically named files
   - ASCII, Binary, XML or ODMG Object Transfer Format physical representation
   - A streaming interface (one similar to the mechanism provided by oobackup would be adequate)
   - A user defined "plug-in" module.

3. The export must not assume that the import will be into a federation with the same page size.

The **ooimport** tool needs to:

1. Allow the recipient of an export set to run a tool that checks whether or not the schemas in the target federation need to be updated and to specify whether or not to allow the import to occur. The user should also be able to choose between object migration modes (immediate, deferred etc.)

2. Allow the recipient to selectively import the exported data. The selection would be based on OIDs, federation/database/container system names and/or object classes. The recipient would perform any content-related selection after importing the data into a temporary federation or databases. This facility would allow remote users to selectively download portions of a very large dataset from a distribution point. [This would have been very beneficial in tiered systems like the ENSO digital music distribution system and the Expeditors freight handling system.]

3. Allow the importer to specify whether or not to try to match objects in the export set with existing objects, identified by system name or OID.

Both tools should checkpoint their processing at reasonable (and possibly user specified) intervals as some of the runs could take many hours. Both tools should be able to handle the largest possible federations without falling prey to cache or filesystem limitations.

Although it is a low priority, it should be possible to give the user (especially the importer) a reasonable projection of the time it will take to process the requested export/import.

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

All of the features would be offered as standard features of Objectivity/DB. It may be possible to modify oobackup and oorestore to accommodate these features, but it is preferable to package them as two extra tools to avoid confusion. It may also be possible to phase out oodump and ooload and replace them with ooexport and ooimport.

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

Users who think they can guarantee identical federations are writing their own utilities or using oocopydb and ooattachdb. This is an incomplete and inefficient workaround. RDBMS users expect to see fast load, import and export tools as a part of the product.

What languages must support this capability?

- C++ and Java
- The SQL SELECT feature will only be available to licensed users of Objectivity/SQL++.

Which platforms must be supported?

- All current platforms, but we could focus on the platforms used by the VLDB sites first.
- The tools must be platform independent.

Do any competitors already have this feature?

- Oracle 8.1, Sybase and Informix.

Customers who require this feature

- Data warehousers – tens of Terabytes per warehouse.
- VARs who ship both schemas and objects to their customers.
- Reconnaissance systems (for Unmanned Autonomous Vehicles and satellites)
- Intelligent battlefield applications
- TRW (Government Agencies) – Terabytes per year
- Lockheed-Martin – tens of Terabytes per year
- CERN, SLAC, SDSS and other HEP/astrophysics sites.

Revenue at risk, or which could be won

- CERN and its collaborators have already generated approximately $1 M. They could easily generate that much again this year.

- We would have a competitive advantage against POET and other embedded DBMS vendors.

When is this required?

- 2001 or Release 7.

Additional Notes

We could consider adding a SORT option to ooexport, but it is probably better to allow use of existing sort utilities.

We will also need:

- Marketing collateral
- Sales and DBA training material

We should consider outsourcing these tools as they can be built using the standard APIs and the Active Schema option. For instance:

- Offer a bonus to our own staff for agreed deliverables (designs, code, QA or documentation)
- Use an external consultancy
- Find a viable bidder via the Internet