Market Requirements Document

Feature Name: Bulk Data Import Tool

Version: 1 **Date Submitted:** 01/12/05

Completed By: Leon Guzenda

Description of the Problem

Users frequently want to move data from external systems on a one time, periodic or triggered basis. Objectivity/DB can import objects in a proprietary format, using a tool named ooload, or in XML format using ooimport. However, these ASCII formats are verbose and slow. For instance, dumping a relatively small federation at one customer site takes over a day. Loading the data back into the federation takes almost two days.

Description of the Requested Feature

This MRD recommends building a bulk data load tool that can accept tabular or delimited records and relationships from an ASCII file and rapidly load it into an Objectivity database or container. The object schema must be pre-defined.

The tool must provide at least the following capabilities:

- The user must be able to define the mapping between the record and field types and the object definitions. By default, the object and attribute names may be used as the record and field names.
- The user must be able to label each record with an integer or alphanumeric key.
- The user must be able to define relationships:
  - With each incoming record, using a record key, or
  - After all records, using record keys, or
  - In a separate file, using record keys.
- Objects may be named in the scope of another object, a container, a database or the federation.
- The user should be able to define how records are clustered.
- The user should be able to create ooMaps and indices during or after loading the actual data.
- The load tool should attempt to co-locate association Varrays with their owning object.
- Floating numbers may be supplied in hexadecimal format to preserve accuracy.

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

This is a new feature.

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

Users must build their own data loaders. The new tool will probably not entirely remove this burden, but it should deal with most of the common cases that we have encountered in the field.

What languages must support this capability?

- C++
- Java.

Which platforms must be supported?

- All current platforms
- Real-time platforms probably do not require this capability. The data can be loaded from a remote processor.

Do any competitors already have this feature?

- DB2, Microsoft SQL Server, Oracle and Sybase.

Customers who require this feature

- VLDB sites.
- Any application that loads data from another system..

Revenue at risk, or which could be won

- Having this tool will improve the chances of being selected for many new applications.

When is this required?

- Release 10.

Additional Notes

We will also need:

- Marketing collateral
- Technical Publications
- Sales training material
- Appendix 1 is an example of an ASCII record format which illustrates one possible format for the input file(s).
- The load tool should read large blocks of the input file to speed up the process.

Appendix 1 – Bulk Load File Formats

There will be at least three kinds of specification:

- Record to object mapping
- Records, optional relationships and optional names
- Relationships
- Object names
- Clustering directives
- OoMap directives
- Indexing directives

A data file might look like this: (the Functional Specification will define the exact syntax)

```
#Mapping
Customer CustObj
Customer.name CustObj._name
Customer.type CustObj._type
Product ProdObj
Product.name ProdObj._name

#Index
Customer.type #DB

// Records

#Type Customer
1:Customer_A, Commercial
2:Customer_B, Aerospace
3:Product_X:bought_by 1,2
4:Product_Y:bought_by 2
5:Product_Z:bought_by 1

#Names
1 My_Name_Is_CA #DB
2 My_Name_Is_CB #DB
5 Best_Seller #FD

#Relationships
1 used_to_buy 5
2 used_to_buy 5
```