

Market Requirements Document

Feature Name: VLDB LongOID

Version: 3 **Date Submitted:** 01/31/05

Version: 2 Date Submitted: 01/25/05

Version: 1 Date Submitted: 01/23/05

Completed By: Leon Guzenda

Description of the Problem

It is possible to access multiple federations in the same program by using a separate session for each, as long as the schemas are compatible. However, there is no simple way to support associations, scalable collections, indices or hash tables that reference objects in more than one federated database.

Description of the Requested Feature

The product should support an additional OID, consisting of a Federated Database ID, Database ID, Container ID, Logical Page Number and Logical Slot Number. It would be known as an LREF (rather than REF or [Short] SREF).

There will have to be a tool and an API for cataloging a federation within another federation.

Multi-federation sessions and multi-federation transactions will be required.

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

Additional kernel capability.

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

Users may be solving this by using structs with an FDID and a REF. However, there is no kernel support for this kind of "OID".

What languages must support this capability?

- All.

Which platforms must be supported?

- Tier 1 platforms, then others as required.

Do any competitors already have this feature?

- No.

Customers who require this feature

- Very Large Database sites, especially multi-site operations.
- SLAC.
- Government customers, such as CSE.

Revenue at risk, or which could be won

- This will increase our lead in the VLDB market.

When is this required?

- Release 10.

Additional Notes

1. We will also need:

- Marketing collateral.
- Updated Technical Publications.
- New QA material.

2. It should be noted that the classes ooLongId and ooLongRef(ooObj), a 96-bit OID, are already defined in Release 9, but there isn't much that can be done with them at this point.