Objectivity Inc.

Backlevel Catalog Handling MRD

Version 1.0

Revision History

Date	Version	Description	Author
12/2/2008	1.0	Initial	Leon Guzenda

Table of Contents

MRD

1.Introduction

The purpose of this document is to collect, analyze and define high-level needs and features of the *Backlevel Catalog Handling* project. It focuses on the capabilities needed by the stakeholders, the target users and why these needs exist. The details of how the *Backlevel Catalog Handling*project fulfils these needs are detailed in the software requirements document (SRD).

1.1.Definitions, Acronyms and Abbreviations

1.2.References

1.3.Overview

The catalog structures within Objectivity/DB remained essentially unchanged until Release 9. The original design used a catalog of databases within the federated database file (database #1) and a catalog of containers within each database. The catalog had a named root object and associated, keyed (by ID) objects representing a database or container. Although lookups by database ID or database name were quick, adding new databases required a scan of the database objects. This did not scale well as users started to create federations with thousands, or tens of thousands, of databases.

At Release 9 the catalogs were completely redesigned to meet new requirements, such as container files, and to make them more scalable. A tool was provided for updating the catalog of databases and the individual catalogs of containers within each database. It is possible for a Release 9+ application to continue using pre-R9 catalogs.

2.Feature Requirements

2.1.Problems

Some users create “master” databases using Release 9+ and then copy them, send them to other sites and attach them to similar federations at those sites. However, if the database has been created with Release 9+ they have the new catalog structure. Although they can be attached to a pre-R9 federation any pre-R9 application is unable to access the containers within the newly attached database. This is causing problems with a major deployment at a government site and the responsible contractor filed an SPR to ask for a product change.

An SE or consultant led optimization usually results in some type of application maintained global collection that leads to containers that then have indexes, but because of P6, P2, and P3, this workaround means that utilization of this global collection must be managed by the application.

2.2.Goals

Make it possible for pre-R9 applications to access databases created by R9+.

As we cannot change the pre-R9 applications we must provide a means to create a database in pre-R9 catalog format with R9+ and then be able to access it with either R9+ or pre-R9 code. It would be acceptable to convert an R9+ catalog to pre-R9 format during a database copy operation.

This means solving PROBLEM32

3.Relationship to other features

3.1.Part of an optional feature?

No.

3.2.Other features required.

Release 9+

3.3.Other features requiring this one.

None.

3.4.Other features that work with this one.

Both pre-R9 and R9+ code must be able to work with pre-R9 databases created with R9+.

4.Positioning

4.1.Competitors that have this capability:

N/A.

4.2.Customers that require this capability:

Northrop Grumman Corporation, Redondo Beach, CA.

4.3.Revenue at risk or which could be won:

This is a revenue generating opportunity as we have a special services contract with NGC.

4.4.When is this required?

This is required as soon as possible.

5.Extent.

5.1.Languages that must support this capability:

C++.

5.2.Platforms that must support this capability:

Solaris.

6.Applicable Standards

None.