Please Note: **Release of this study by the USDA Forest Service does not constitute an endorsement of the accuracy of the study, or an endorsement of the findings or recommended solutions for integrating/interfacing ICBS and ROSS.**

-Andy Gray
ICBS-R Project Manager
June 28, 2004

Resource Ordering and Status System (ROSS) Program

National Interagency Incident Cache Business System (ICBS) Engineering Study

August 25, 2000

TABLE OF CONTENTS

1. INTRODUCTION 1

1.1 PURPOSE 1
1.2 OVERVIEW 1
1.3 BACKGROUND/SCOPE 2
1.4 TECHNICAL ASSUMPTIONS 3
1.5 ROM ESTIMATE ASSUMPTIONS 3

2. TECHNICAL CONSIDERATIONS 5

2.1 GENERAL TECHNICAL CONSIDERATIONS 5
2.2 ROSS/ICBS COMMON RDBMS LEVEL TECHNICAL INFRASTRUCTURE 5
2.2.1 Advantages of Upgrading from Oracle 7.3.4 to Oracle 8.1.x 5
2.3 TECHNICAL APPROACHES REJECTED 7
2.3.1 Manual Dual-System/Dual-Entry of Information 7
2.3.2 File-Based Data Sharing between ROSS and ICBS 7
2.3.3 Semi-Automated Transfer of Data 7
2.3.4 Oracle-Specific Mechanisms 7
2.4 PREMISES FOR PROPOSED ALTERNATIVES 8

3. DATABASE INFRASTRUCTURE TECHNICAL OPTIONS 9

4. DATABASE STRUCTURE TECHNICAL OPTIONS 11

4.1 OVERALL ENGINEERING ISSUES 11
4.1.1 Primary Keys and Unique Indexes 11
4.1.2 Foreign Keys and Referential Integrity 11
4.1.3 Triggers on Replicated Tables 11
4.1.4 Data Types 11
4.1.5 Time 11
4.1.6 Sequences 12
4.2 ORACLE ROW-LEVEL SECURITY, INSTALLATION AND MAINTENANCE REQUIREMENTS 12
4.2.1 Security 12
4.2.2 Updateable Snapshots 13
4.2.3 Multi-Master Replication 14

5. APPLICATION-LEVEL TECHNICAL OPTIONS 17

5.1 LOCATION FOR "APPLICATION-LEVEL" CHANGES 17
5.2 APPLICATION/DATA STRUCTURE ALTERNATIVES 17
5.2.1 Data Translation 18
5.2.2 Partial Integration 18
5.2.3 Full Integration 18
5.3 ASSESSMENT OF ICBS FUNCTIONALITY 18
5.4 APPLICATION-LEVEL TECHNICAL OPTION CONSIDERATIONS 24
5.4.1 Considerations for Partial Integration and/or Full Integration 24
5.4.2 Considerations for Data Translation 24
5.4.3 Hybrid Approach 24

6. HARDWARE INFRASTRUCTURE TECHNICAL OPTIONS 25

6.1 PRIMARY PLATFORM CONSIDERATIONS 25
6.2 PLATFORM ALTERNATIVES 26
6.2.1 Minimum Configuration 26
6.2.2 Partially Redundant Configuration 27
6.2.3 Fully Redundant (CPU Level and Data Level) 28
6.2.4 Fully Redundant Modularly Expanded Configuration 29

6.3 HARDWARE-LEVEL RECOMMENDATIONS 31

7. NETWORK INFRASTRUCTURE TECHNICAL OPTIONS 32

8. PROPOSED ALTERNATIVES 33

8.1 NO COST ALTERNATIVE 34
8.1.1 Description/Rationale 34
8.1.2 Database Impact 34
8.1.3 Application Impact 34
8.2 ALTERNATIVE #1, UPDATEABLE SNAPSHOTS/TRANSLATION 35
8.2.1 Description/Rationale 35
8.2.2 Database Impact 35
8.2.3 Application Impact 36
8.3 ALTERNATIVE #2A, ICBS MULTI-MASTER/UPDATEABLE SNAPSHOTS 37
8.3.1 Description/Rationale 37
8.3.2 Database Impact 37
8.3.3 Application Impact 38
8.4 ALTERNATIVE #2B, ICBS CONSOLIDATION/UPDATEABLE SNAPSHOTS 39
8.4.1 Description/Rationale 39
8.4.2 Database Impact 39
8.4.3 Application Impact 40
8.5 ALTERNATIVE #3A, IDENTICAL SCHEMAS/MULTI-MASTER 41
8.5.1 Description/Rationale 41
8.5.2 Database Impact 41
8.5.3 Application Impact 42
8.6 ALTERNATIVE #3B, IDENTICAL SCHEMAS/UPDATEABLE SNAPSHOTS 42
8.6.1 Description/Rationale 42
8.6.2 Database Impact 43
8.6.3 Application Impact 43
8.7 ALTERNATIVE #3C, INTEGRATED ROSS/ICBS DATABASE 43
8.7.1 Description/Rationale 43
8.7.2 Database Impact 44
8.7.3 Application Impact 44

9. ASSESSMENT OF ADVANTAGES/DISADVANTAGES 45

10. RECOMMENDATIONS 47

10.1 COMPARISON OF ALTERNATIVES 47
10.2 RECOMMENDED APPROACH 49
10.2.1 Phase I 49
10.2.2 Phase II 49
10.2.3 Phase III 50

TABLES

Table 3-1 Oracle Row-Level Replication Options 9
Table 5-1 Impact of Integration Alternative on ROSS/ICBS Application Levels 18
Table 5-2 Summary of Requested Capability in Terms of ICBS Functions 19
Table 5-3 ICBS Order / Fill Related Functions 20
Table 5-4 All ICBS Related Functions 21
Table 6-1, ROSS / ICBS Platform Considerations 25
Table 6-2 ICBS Hardware Level Recommendations 31
Table 6-3 ROSS Hardware Level Recommendations 31
Table 9-1 ROSS/ICBS Integration Options - Disadvantages and Risks 45

FIGURES

Figure 6-1 Minimum Configuration 27
Figure 6-2 Partially Redundant Configuration 28
Figure 6-3 Fully Redundant Configuration (with Improved Data Storage) 29
Figure 6-4 Fully Redundant Modularly Expanded Configuration 30
Figure 8-1 No Cost Alternative – Database Links 34
Figure 8-2 Alternative #1 - Updateable Snapshots 35
Figure 8-3 Alternative #2a - ICBS Multi-Master/Updateable Snapshots 37
Figure 8-4 Alternative #2b - ICBS Consolidation/Updateable Snapshots 39
Figure 8-6 Alternative #3a - Identical Schemas/Multi-Master 41
Figure 8-7 Alternative #3b - Identical Schemas/Updateable Snapshots 42
Figure 8-8 Alternative #3c - Integration ROSS/ICBS Database 44
Figure 10-1 ROSS/ICBS Integration Alternatives (Risk versus Cost) 47
Figure 10-2 ROSS/ICBS Integration Alternatives (Disruption to Database and Application Environments vs Level of Database Integration) 48
Figure 10-3 Recommended Phased ROSS / ICBS Approach 50

APPENDICES

<Appendix A and B have been removed from this version>

APPENDIX C. ACRONYMS **1**

APPENDIX D. DEFINITIONS **1**

1. Introduction

1.1 Purpose

The purpose of this National Interagency Incident Cache Business System (ICBS) study is to identify, develop and assess valid technical alternatives that optimize integration of the ICBS and the National Interagency Resource Ordering and Status System (ROSS).

1.2 Overview

Many considerations must be balanced in trying to assess technical alternatives for sharing information between two complex applications. As might be expected, some of the most critical decisions must be made based upon funding, functional, operational, organizational, and political issues rather than purely technical factors.

Such "non-technical" considerations would include:

- Levels of available funding for development
- Priorities assigned to integrated ROSS/ICBS functional capabilities
- Required deployment schedules for high-priority functions
- Political, functional, and operational reasons for allowing physically dispersed databases to remain dispersed
- Political, functional, and operational reasons for consolidating physically dispersed databases
- Overall operational maintenance and support strategies and funding
- Overlapping and parallel development efforts
- Strategic long-term goals and objectives

This ROSS/ICBS engineering study is premised upon the belief that each proposed approach for the integration of ROSS and ICBS must be capable of modular implementation and must be consistent with an overall strategy for the comprehensive integration of the ROSS and ICBS systems. However, depending upon the "non-technical" considerations driving the decision-making process, the initial approach, near-term, mid-term, and long-term target states can vary greatly.

Any single ROSS/ICBS integration solution will involve engineering and design approaches with respect to:

- Database Infrastructure Technical Options (See Section 3)
- Database Structure Technical Options (See Section 4)
- Application Level Technical Options (See Section 5)
- Hardware Infrastructure Technical Options (See Section 6)
- Network Infrastructure Technical Options (See Section 7)

Decisions made within each of the technical areas listed above limit the viable alternatives within each successive technical area. For example, "non-technical" considerations that drive the selection of Database Infrastructure Technical Options will have implications in all successive areas.

The intent of this ROSS/ICBS integration study is to present truly viable technical alternatives for integrating ROSS and ICBS in an unbiased, technical manner. Each proposed alternative proposed is consistent with a single integrated ROSS/ICBS vision, even though full integration of ROSS and ICBS may

not be the final objective of the customer. The most basic alternative is intended to serve as an initial platform that can be modularly enhanced to provide varying degrees of capability within the bounds of that alternative. Each alternative is compatible with and is capable of transformation to the next successive alternative without loss of prior development.

1.3 Background/Scope

The ROSS and the ICBS systems are both complex systems that use application unique data structures and graphical user interfaces (GUIs) to support very specific user community objectives. The objective of the ROSS system is to support the location, allocation and de-allocation of critical resources to forest wildfire-fighting operations and to similarly dangerous and potentially life-threatening local, state and national operations. The objective of the ICBS is to support the maintenance of National Interagency depots that house and maintain equipment and supplies for use in the same local, state and national operations supported by the ROSS system.

A large volume of the resources allocated by the ROSS system is received from the National Interagency depots supported by ICBS. As a result, a significant overlap exists with respect to the information maintained by the two systems. Such duplication leads to operational inefficiency due to the need to coordinate and enter redundant information manually. Furthermore, delays in updates and human errors can introduce inconsistencies between the two information stores.

The scope of this engineering study is to identify, develop and assess **valid** technical alternatives for the integration of ICBS and ROSS information and operations. The categories for the required range of alternatives included the following:

- Current situation as identified in the ROSS Build Phase contract, referred to as the "No Cost" alternative.
    - Read NFES Catalog Information from ICBS
    - Read Cache Inventory Information from ICBS
- Implementation of the following interfaces with no re-hosting/re-engineering of ICBS
    - Read NFES Catalog Information from ICBS
    - Read Cache Inventory Information from ICBS
    - Designate items as critical within the NFES Catalog Data within ROSS
    - Transmit and receive supply order/fill information between ROSS and ICBS
- Implement all interfaces listed above through re-host/re-engineering of ICBS, including integration as required for the sharing of common data.

While every attempt has been made to present alternatives within each of these categories, it is not possible to satisfy the interface requirement for transmitting and receiving supply order/fill information between ROSS and ICBS without partially re-engineering the ICBS system. Consequently, all of the proposed alternatives involve re-engineering of ICBS to some degree.

The technical goal is to improve the operational capabilities and increase the reliability of both systems through the sharing of data. As a result, this document does not provide equal analysis time to all "potential solutions." If a potential alternative is believed to be invalid, the rationale for this conclusion is presented and no further consideration is presented for that alternative. The technical alternatives covered within this study are as follows:

1 – No Cost

2 – Updateable Snapshots / Translation

3 – ICBS Consolidation / Updateable Snapshots

4 – ICBS Multi-Master / Updateable Snapshots

5 – Identical Schemas / Updateable Snapshots

6 – Identical Schemas / Multi-Master

7 – Integrated Database

Within the alternatives listed above, several technical aspects were evaluated, including:

- Oracle Row-level Replication
- Row-Level Replication (Installation & Maintenance)
- Impact of Integration on ROSS / ICBS Application Levels
- Overlapping ROSS / ICBS Functionality
- ROSS / ICBS Platform Configurations
- Network Infrastructure Options

1.4 Technical Assumptions

The following assumptions were used to guide the selection of alternatives and the recommended approach:

- Read/Write capability with respect to both ICBS and ROSS data is required.
- ICBS data must be accessible from the ROSS GUI for the applicable ROSS functionality.
- ROSS data must be accessible from the ICBS GUI for the applicable ICBS functionality.
- Operational concerns may require the ICBS and ROSS repositories to remain autonomous.
- Initial development dollars may be limited.

1.5 ROM Estimate Assumptions

In preparing the ROM estimates for each of the proposed alternatives, certain key assumptions were made. This section describes these key assumptions. **<ROM ESTIMATES ARE NOT INCLUDED IN THIS VERSION>**

Configuration of the ICBS database and application

It is assumed that the developer of any alternative proposed in this study will be an active participant in the ICBS database and application configuration management process. Each of the proposed alternatives is dependent on having an ongoing, complete and thorough understanding of the ICBS database structure and application. Ideally, and in accordance with standards and procedures, ICBS database and application baselines will be established prior to initiation of the ROSS/ICBS detailed design process. The ROM

estimates for each of the alternatives has been prepared assuming minimal change to either of these two baselines during the design, implementation, and deployment phases of the life cycle. Additionally, it is assumed that the database schemas of the individual ICBS databases are equivalent.

Data Quality and Integrity

Data quality and integrity is the responsibility of the government. The developer can only guarantee that the data structures are consistent with and support the underlying business rule requirements and operate as specified to meet these requirements and cannot guarantee the quality and/or integrity of the data.

Hardware and Network Infrastructure

It is assumed the current network infrastructure will provide sufficient communication bandwidth between the two systems hosted on their respective networks. Additionally, it is assumed that the developer will be provided network configuration and maintenance support on an as needed basis.

Access to Government Subject Matter and Functional Experts

It is assumed that developer will have ready access to ICBS subject matter and functional experts. Continuous access to these key personnel will help ensure successful results.

2. Technical Considerations

2.1 General Technical Considerations

Numerous considerations must be balanced in attempting to assess the many and varied technical alternatives for sharing information between two complex applications. Although no approach will be optimal, an "acceptable" alternative must possess the following general characteristics:

- Meet initial operational requirements
- Allow modular expansion of operational capability in accordance with a long-term plan
- Minimize initial development and implementation costs and risks
- Meet or exceed existing operational reliability requirements
- Assure compatibility with existing technical infrastructure
- Leverage existing or available COTS capabilities rather than recreating capabilities that can be acquired through COTS resources
- Minimize disruption of existing systems

2.2 ROSS/ICBS Common RDBMS Level Technical Infrastructure

The existing ICBS system is based upon an Oracle 7.3 database. Users access and manipulate information within this database via a client application developed using Oracle*Forms, RDBMS level triggers that perform server-side processing, and Oracle-stored procedures. As such, the ICBS system is tied to an Oracle 7.3 compatible RDBMS. The ROSS system server-side application code and client-side applications were developed in a Java/CORBA-based development environment using a product called Versata Studio that is "RDBMS independent." The current ROSS deployment configuration uses an Oracle 8.1.6 RDBMS. Because transition from Oracle 7.3 to Oracle 8.1.6 is fully forward compatible, the ICBS could be migrated to Oracle 8.1.6 with minimal development and implementation costs.

2.2.1 Advantages of Upgrading from Oracle 7.3.4 to Oracle 8.1.x

It is strongly recommended that the ICBS upgrade to the latest release of Oracle. Oracle8 and 8i offer significantly enhanced replication features that are not available in the 7.3.4 release of the database. These features are so significant that it would be wise not to create a replicated environment until this upgrade occurs. The following is a summary of the new functionality offered by major release:

Oracle8 Enhancements:

- Parallel Propagation: Oracle8 can deliver deferred transactions to remote databases in multiple streams, thereby significantly improving throughput.
- Internalized Triggers: A number of major triggers supporting replication has been internalized i.e. they are written in C and have been compiled directly into the database engine. This results in faster execution and generation of replication support.
- Reduced Data Propagation: With Oracle8 one can omit columns in a table from replication. What this means is that the replication facility does not check the before and after values of the columns that you so designate. Since these columns are not replicated, less data is transmitted, and less time is spent checking for conflicts.

- Subquery Snapshots: Snapshots containing subqueries can be fast-refreshed provided that they meet certain restrictions.
- LOB Support: Oracle8 supports the replication of LOB, CLOB, and NCLOB datatypes (which are new to Oracle8). The Oracle7 predecessors of these datatypes, LONG and LONG RAW could not be replicated.
- Fine-Grained Quiesce: Oracle8 can quiesce one replication group at a time. In Oracle7, quiescing one group effectively quiesces all other groups.
- Primary Key Snapshots: Oracle8 snapshots are based on the primary key, rather than on ROWID. Thus, master tables can be reorganized without having to perform complete refreshes on all snapshot tables.
- Master Site Snapshot Registration: When one creates a snapshot from an Oracle8 snapshot site to an Oracle8 master site, Oracle records the existence of the snapshot in the master site's data dictionary view DBA_REGISTERED_SNAPSHOTS.
- Support for Offline Instantiation: The procedure for adding new master and snapshot sites requires less downtime and is more automatic.
- Deferred Constraint Checking for Updateable Snapshots: Uniqueness and referential integrity constraints on updateable snapshots can be checked and enforced after a refresh is complete instead of during the refresh.
- Validation Procedure: Oracle8 provides the DBMS_REPCAT.VALIDATE procedure which can help to verify the correct configuration of a replicated environment.
- Partitioned Tables and Indexes: Oracle8 supports the replication of partitioned tables and tables with partitioned indexes.
- Global Users and Global Roles: Oracle8 provides a user management scheme that supports maintenance of users and roles across multiple database instances. Instead of having to visit every instance to grant privileges, create users, and so on, one can define users and roles in such a way that changes from a central location take effect everywhere.
- System Security Model: The management of users in an advanced replication environment is simplified tremendously in Oracle8, with the introduction of propagator and receiver accounts. Instead of having to create a user in all instances participating in the replication and having to create and verify private database links for each user, you can designate one account to queue DML and one account to apply DML.

Oracle8i Enhancements:

Oracle8i enhancements to replication focus primarily on improved performance, simpler administration, and mass deployment (e.g. hundreds of updateable snapshots from a single master table)

- Snapshot Refreshes have been optimized to support refresh groups with up to 400 snapshots per group. In addition, the refresh algorithm has been modified to significantly reduce the number of round-trips required to perform a refresh.
- Snapshot deployment templates allow the DBA to control the contents of snapshot sites. The template are defined at the master site and deployed to snapshot sites, as opposed to snapshot sites defining their own snapshots. This ensures a uniform configuration of all snapshot sites.

- Vertical Partitioning of Updateable Snapshots: Oracle8i provides the capability to vertical partition updateable snapshots. However, vertical partitions must include all columns that comprise the master table's primary key, and the columns that are not included must either be nullable in the master table or have default values assigned.

2.3 Technical Approaches Rejected

ROSS and ICBS are both real-time dispatch systems. Both serve critical functions supporting life-threatening situations. To meet their objectives, both need access to timely and accurate resource availability information. For the reasons indicated below, the following technical approaches have been summarily rejected.

2.3.1 Manual Dual-System/Dual-Entry of Information

This approach represents the status quo for ROSS and ICBS users. It consists of ICBS users using both the ROSS and ICBS systems, making manual, redundant entries in each system to maintain synchronization. This alternative has been rejected for the following reasons:

- Overly burdensome to the user community
- High risk of incomplete, inaccurate, unreliable synchronization of information
- Potential life-threatening impact to ROSS operations due to failure to provide accurate, complete, and reliable information

2.3.2 File-Based Data Sharing between ROSS and ICBS

One means of sharing information between systems is to develop an application-level process by which changes in the respective systems are identified and placed into discrete files, such as XML-based text files with associated data table definition (DTD) files. These files are parsed by the systems sharing information and used to update their respective data stores. Such a file-based approach is extremely useful to interface with functionally and organizationally independent applications and legacy applications. While this approach could be used to meet short-term ROSS/ICBS requirements, it is best used to support periodic, manual or semi-manual transfers of complete sets of information, but is not designed to synchronize large databases on an automatic, near real-time basis. For this reason, this approach is believed to be inadequate for long-term use by ROSS and ICBS and should be avoided. Furthermore, existing RDBMS COTS capabilities can meet initial ROSS/ICBS data-sharing requirements as well as long-term ROSS/ICBS data-sharing requirements with less development costs. If one attempted to develop a file-based system that was capable of synchronizing components of large databases in an automatic, near real-time basis, in effect one would be building an inferior instantiation of the existing RDBMS COTS capabilities.

2.3.3 Semi-Automated Transfer of Data

This approach consists of the automated transfer of application specific information between ROSS and ICBS based upon user-initiated actions within each of the applications. This alternative requires the same data translation software development required by more comprehensive solutions, and necessitates significant ROSS/ICBS application and GUI modifications, yet yields inferior capability compared to fully automated approaches based upon Oracle COTS data replication features.

2.3.4 Oracle-Specific Mechanisms

The following Oracle specific mechanisms have also been rejected as potential RDBMS level technical alternatives: {Note: Although "database links" have been rejected as a modularly expandable RDBMS level

technical approach, the "no cost" alternative does propose the use of "database links" to provide read-only access in support of report generation.}

- Export/import utility represents the most primitive method of sharing data among Oracle databases. This utility offers a static means of transferring data. It does not support active synchronized transfers of data between databases.
- Database links provide transparency to remote objects in other databases. Except in instances in which only a very few objects are being replicated, this option by itself would not provide the robustness needed for actively sharing data between remote databases. The specific reasons for why this is true include:
  - Automated refresh of data: Oracle's built-in replication features guarantee automated replication of table data to remote databases, Oracle data links do not
  - Propagation of Data Changes back to Source Tables: Oracle's built-in replication features automate the propagation of DML back to the source tables, Oracle data links do not
  - Transaction Consistency: Oracle's built-in replication preserves the order of transactions when it replicates them to participating sites, and it also guarantees the consistency of data at all sites participating in the replication environment, Oracle data links do not
  - Conflict Avoidance and Resolution: Oracle's built-in replication features provide built-in conflict avoidance and resolution methods, Oracle data links do not

  In short, the use of database links would require the development of significant custom code to address and handle all of the issues described above which are included in Oracle's built-in replication features.

- Read-Only Snapshots, where a master table is copied to one or more databases. Changes in the master table are reflected in the snapshot tables whenever the snapshot refreshes. The snapshot site determines the frequency of the refreshes. The data at the snapshot site can be used by other applications, but no updates to the master table are made based upon changes made by such applications to the snapshot site database.

2.4 Premises for Proposed Alternatives

The following premises serve as the basis for the alternatives proposed by this engineering study:

- Each alternative must be capable of modular implementation and must be consistent with an overall strategy for the comprehensive integration of the ROSS and ICBS systems.
- Each alternative is consistent with a single integrated vision at the data level, even though full integration of ROSS and ICBS may not be a near-term objective of the customer.
- Each alternative is intended to serve as an initial platform that can be developed and implemented as an increment of capability within the overall ROSS/ICBS integration strategy.
- Each alternative must be compatible with and be capable of transformation to the next successive alternative without loss of prior development.
- The alternatives proposed in this study are based upon varying levels of Oracle replication, varying levels of ROSS/ICBS data structure redesign, varying levels of ROSS/ICBS data transformation, and varying levels of ROSS/ICBS application level redesign.

- All proposed alternatives support migration towards the establishment of a single, integrated ROSS/ICBS data structure that requires no data replication for both applications to use and share information from a single centralized data store.

3. Database Infrastructure Technical Options

Two types of replication were considered: row-level and procedural. Oracle row-level replication includes basic replication (or read-only snapshots), updateable snapshots, and multi-master replication. Procedural replication, on the other hand, replicates calls to PL/SQL-packaged procedures and functions. For on-line transaction processing (OLTP) workloads, row-level replication makes the most sense. There is no real benefit to writing PL/SQL packages to perform all of an application's DML. Furthermore, row-level replication offers conflict resolution, whereas procedural replication does not. Oracle row-level replication was consequently chosen as the basis for the proposed alternatives for the ICBS Engineering Study.

- The capability to replicate data and DDL from one database to one or more databases **automatically**
- The capability to maintain transaction consistency, thus preserving the order of transactions when the transactions are replicated to participating sites in the replicated environment
- The built-in capability to guarantee the consistency of data at all sites
- Advanced, or multi-master replication capability that provides near real-time data exchange between databases
- Oracle's replication capabilities built into the database itself, thus requiring no development effort to implement replication

Each of the proposed alternatives includes one or a combination of the three types of Oracle row-level replication. Table 3-1 describes the various types of Oracle row-level replication and their associated restrictions.

Table 3-1 Oracle Row-Level Replication Options

Replication Type	Description	Restrictions
Updateable Snapshots	A master table is copied to one or more databases. Changes in the master table are reflected in the snapshot tables whenever the snapshot refreshes. The snapshot site determines the frequency of the refreshes. The snapshot sites are able to modify the data and send their changes back to the master. The snapshot site determines the frequency of the refreshes and the frequency with which updates are sent back to the master.	Updateable snapshots must be simple snapshots. That is, they must be snapshots against a single table without DISTINCT, GROUP BY, or CONNECT BY operators or any sub-queries. Columns of type LONG and LONG RAW cannot be used. Updateable snapshots must include all columns of the master table.

Replication Type	Description	Restrictions
Multi-Master Replication	A table is copied to one or more databases, and each database has the ability to insert, update or delete records from it. Modifications are pushed to the other databases at an interval that the DBA sets for each replication group. The highest theoretical frequency is once per second.	There is no support for cascading deletes. As an alternative, one may consider writing a trigger on the master table to delete child records. Sequences do not replicate. If you use sequences to populate key fields, be sure to designate a range of sequence values in each master database that is large enough to avoid key collisions for the life of the application. There is no support for local customization of replicated tables. In other words, replicated tables must have an identical shape in each master database. The data types LONG, LONG RAW, and HHCODE do not replicate. You may replicate tables containing columns of these data types, but DML to these columns will not propagate to other master sites. A recommendation is to use the CLOB and LOB data types available in Oracle 8.

4. Database Structure Technical Options

This section addresses database structural technical issues that are pertinent to implementing any of the proposed alternatives. Section 4.1 describes overall database engineering/structural issues that are common to each of the Oracle row-level replication types that serve as the basis for each of the proposed alternatives. Section 4.2 presents the security, installation and maintenance requirements for implementing each type of row-level replication

4.1 Overall Engineering Issues

4.1.1 Primary Keys and Unique Indexes

Every replicated table must have either a primary key or a unique index. Ideally, primary keys should consist of a single column, preferably with numeric values. Whenever Oracle must locate a row, which occurs quite frequently with row-level replication, the task should be as fast as possible. Using a single field primary key helps the overall performance of the replication functions. The primary advantage to using fields over VARCHARs for primary keys is that the indices on numeric columns are generally much smaller than those on VARCHARs. However, as will be explained in Section 4.1.6, the recommendation for each of the proposed alternatives will be the use of a multicolumn primary key.

4.1.2 Foreign Keys and Referential Integrity

Foreign key constraints are supportable and even advisable in a replicated environment. Oracle 8 allows primary keys and unique indexes on snapshot base tables. Oracle 8 can even create a primary key on snapshot tables automatically.

4.1.3 Triggers on Replicated Tables

An analysis is needed of the ICBS and ROSS database environments to ensure that application-related triggers do not interfere with replication functionality. All of Oracle's replication triggers are after-row triggers. Because it is not possible to control the firing order of triggers of the same type, it is best to guarantee that the application-related triggers fire before the replication triggers. Controlling the occurrence of triggers requires making the application-related triggers fire before row-level triggers.

4.1.4 Data Types

Oracle does not replicate LONG or LONG ROW data types. Oracle will replicate a table that has a LONG or LONG ROW column, but changes to the values in such columns are not propagated. Oracle 8 addresses this limitation by supporting replication of CLOB (character large object) and BLOB (binary large object) data types. Nevertheless, applications should perform minimal updates to these data types in a replicated environment because these actions can adversely impact the network.

4.1.5 Time

Among the most common and easy to use methods of conflict resolution in a replicated environment are Latest Timestamp and Earliest Timestamp. To utilize these fields, replicated tables should have a timestamp field and a before-row-trigger to populate the table on every insert and update. For this field to be effective, the system clocks on the machines hosting the replicated databases must be synchronized. Synchronization is particularly important for applications that perform at high transaction rates, especially if these transactions are not partitioned to avoid conflicts.

The time zone of each database server may also be an issue. If time zone biases may be an issue, it is advisable either to standardize to a single time zone, such as Greenwich Mean Time (GMT), or to adjust the value of the timestamp when the field is populated. The former strategy is much simpler but may cause confusion and complications if the application presents time-sensitive data to the user.

4.1.6 Sequences

Oracle's advanced replication facility does not replicate sequences, which are often used to generate primary key values and other numeric keys. Because of this limitation, applications that use sequences can reference a single sequence that is located at a master site, can use their own local sequence, or can use a multicolumn primary key. It is recommended in the proposed alternatives that a multicolumn primary key be used. This primary key will consist of two columns. The first column will be an alpha character that will be used to identify the site. The second column will be a sequence number. The advantages of this method are (1) that the primary key values are not artificially associated with sites and (2) that there is no danger of running out of sequence numbers at any site. Additionally, new sites can be easily added.

4.2 Oracle Row-Level Security, Installation and Maintenance Requirements

The following discussion presents information about the security, installation and maintenance requirements for implementing each type of row-level replication.

4.2.1 Security

The issues pertinent to implementing security in a replicated environment are common to each type of row-level replication. There are two primary security issues with which to be concerned. The first issue is privilege management (that is, the way access to objects in remote databases is managed). The second issue involves the authentication method used to allow users access to the database itself.

4.2.1.1 Privilege Management

There are four primary choices for managing access to objects in remote databases:

- The simplistic approach allows access to remote objects over a public database link, with a local public synonym for each remote object.
- The mirrored account approach allows access to remote objects over private database links for all user accounts, with a local public synonym for each remote object.
- The local view approach involves creating a local view for remote objects through which remote objects are accessed.
- The local wrapper approach calls remote PL/SQL objects (procedures and packages) from local procedures, which themselves are not available to local users.

For all of the proposed alternatives, it is recommended that the local view approach be chosen to manage access privilege to remote objects in the database, coupled with the local wrapper approach for managing access to remote PL/SQL objects (procedures and packages). The local view approach is being recommended for managing access to remote objects for the following reasons:

- The local view approach solves the problem of controlling access to remote objects with local roles. As many roles as necessary can be created to provide appropriate levels of access.
- There is no public database link involved and therefore no open door to the remote database. The local account that owns the database link and the view does not require CREATE SESSION privileges once the view is in place.

The local wrapper approach is being recommended for these reasons:

- Just as a local view of a remote object facilitates local privilege administration over remote tables, a local wrapper facilitates local privilege administration over remote procedures and packages.
- Additionally, the wrapper can help to ensure data consistency by performing edit checks and setting parameter values.

4.2.1.2 Authentication Methods

Three methods can be used for authenticating users:

- Database Authentication: This method corresponds to accounts made with the CREATE USER command. Users must provide a valid username/password, which the database validates with information stored in the data dictionary.
- Operating System Authentication: These Oracle accounts correspond to operating system accounts. If a user can log in to the operating system, he/she is permitted to log in to the database. This type of account is referred to as OPS$ because the corresponding database usernames are in the form OPS$os_user_name by default.
- External Authentication: These accounts are validated by some external means, such as fingerprint scanning, or by a network-authentication mechanism, such as Kerberos.

Database authentication is recommended as the means for authenticating users of the database. For additional protection to safeguard against the possibility of a user who can obtain arguments passed by other users on the system (e.g., usernames and passwords), the wrapper command around SQLplus is recommended to prevent arguments from being displayed.

4.2.2 Updateable Snapshots

4.2.2.1 Installation

The following conditions and constraints apply to the installation of Updateable Snapshots:

- Updateable snapshots require components at both the master site and the snapshot site.
- Replication administrator accounts exist with proper privileges at the master and all snapshot sites.
- Database links must be in place. The links from the snapshot site to the master site must connect to an account that either is the owner of the master table or has replication administrator privileges. In addition, the account at the master site must have EXECUTE privileges on the package SYS.DBMSOBJGWRAPPER.
- The table to be replicated exists at the master site and has a primary key defined.
- The account that creates the updateable snapshot must have the privileges CREATE SNAPSHOT, CREATE TABLE, CREATE TRIGGER, and CREATE VIEW.
- If the snapshot is to be created in a different schema (i.e., owned by a different Oracle account from the one issuing the CREATE SNAPSHOT statement), the account must have the CREATE ANY SNAPSHOT privilege.
- Preparing the Master Table: An important difference between read-only snapshots and updateable snapshots is that the master table for the latter must be defined as a replicated object. The master table must also have a snapshot log.

- Preparing the Snapshot Site: At the snapshot site, we must first create the actual snapshot, either with the CREATE SNAPSHOT statement or by supplying the DDL in the call to DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT.
- Also, in the CREATE SNAPSHOT statement, one does not specify a NEXT time for the refreshes. This statement is omitted because one puts the updateable snapshot into a refresh group that controls the refresh schedule.
- Updateable snapshots must contain every field in the master table.

4.2.2.2 Maintenance

Snapshots require a certain amount of DBA attention in order to keep them running optimally. Oracle provides packaged procedures to support the DBA. The DBMS_REFRESH and DBMS_SNAPSHOT built-in packages include a variety of routines the DBA can use to manage snapshots and snapshot logs.

Occasionally, a DBA must reorganize a master table (i.e., coalesce its extents and reduce row chaining). There are 2 new modules available in Oracle 8 that allow one to reorganize a master table without invalidating its snapshot log. Therefore, one does not have to perform complete refreshes of the table's snapshots after it is reorganized

In addition to the maintenance issues associated with Read Only Snapshots, the following are maintenance tasks specific to Updateable Snapshots:

- **Altering the Master Table:** If the structure of the updateable snapshot's master table changes, the updateable snapshot must reflect the modification. Since updateable snapshots are registered as replicated objects, a change to a master table will generate appropriate DDL calls for all snapshot sites. However, unlike DDL changes in a multi-master replicated environment, the snapshot site must request the DDL changes from the master
- **Dropping a Replicated Snapshot Object:** One may wish to drop a snapshot, either because it is no longer required at the snapshot site or because the master table no longer exists. Oracle supplies built-in packages to handle this
- **Transaction Management:** DBA needs to carefully monitor the size i.e. total # of records per transaction to ensure that updates do not overwhelm the system. If the volume of particular updates become significantly large to the extent that I/O performance becomes an issue, the DBA may want to consider using procedural replication

Monitoring the additional workload a distributed environment demands of a DBA can be considerable. In addition to the normal DBA responsibilities, such as monitoring space utilization and extent allocation, the DBA must monitor objects such as snapshot logs, job queues, transaction queues, and error queues. If left unresolved, problems in a distributed environment can become so difficult to solve that it is easier to reload data from scratch than try to resolve specific errors

4.2.3 Multi-Master Replication

4.2.3.1 Installation

The following conditions and constraints apply to the installation of Multi-Master Replication:

- Preliminary Steps
  - Determine and set the appropriate initialization parameters
  - Size rollback segments and redo logs

    - Run CATPROC.SQL and CATREP.SQL
    - Create administrative accounts with appropriate privileges
    - Create the necessary database links
- Quick Setup
    - The administrator creates an object in all master databases; a table can be populated with identical data at all locations before configuring it for replication, or the administrator can let Oracle's replication packages populate the table.
    - The administrator creates one or more replication groups using the package procedure. DBMS_REPCAT.CREATE_MASTER_REPGROUP.
    - The administrator adds objects to the replication groups using the package procedure DBMS_REPCAT.CREATE_MASTER_REPOBJECT.
    - The administrator generates replication support for each object using the package procedure DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT.
    - The administrator adds one or more master sites to the replication group using the package DBMS_REPCAT.ADD_MASTER_DATABASE.
    - The administrator schedules the propagation of DML changes and replicated procedure calls to each master site using DBMS_DEFER_SYS.SCHEDULE_PUSH.
    - The administrator schedules the purging of the DEFTRAN queue using the DBMS_DEFER_SYS.SCHEDULE_PURGE.

Additional issues include the following:

- **Replication Groups:** a replication group is a collection of one or more objects, generally tables, that are logically related and that can or should be administered together. As a general guideline, one can place all replicated objects associated wit a given application into a single group. A common practice is to create all of an application's objects under a single schema.
- **Naming Conventions:** Oracle replication has many different entities associated with it: replication groups, column groups, site priorities. Administration is simplified if such objects are named according to a uniform convention. With this end in mind, recommend naming replication groups in the form RG_name.

In the majority of cases, the easiest strategy is associate a single replication group with a single schema, with the following caveats:

- Do not allow transactions to cross replication group boundaries
- Do not allow referential integrity constraints to cross replication group boundaries
- Identify and isolate problem tables into separate replication groups

The following are the most important items to monitor in a multi-master replicated environment:

- Conflict avoidance and resolution
- The number of transactions queued at each originating site
- The number of unresolved errors at each destination site: if one allows errors to accumulate at destination sites, it can be very difficult to clear them

- The number of entries in each snapshot log; if snapshot logs have been created on master tables, one should check the size of the snapshot logs periodically to ensure that they are not growing too large. If a snapshot log has many entries (i.e. thousands), it may be because either not all snapshots that are mastered to the master table are firing or because their refresh interval is too long

5. Application-Level Technical Options

Various levels of integrated ROSS/ICBS capability may be presented as:

- Separate ROSS/ICBS databases as they exist today with data level sharing through translators
- Consolidation of individual databases with data level sharing through translators
- A fully integrated ROSS/ICBS database.

The choice is a matter of preferred development environments and user community preferences that can be addressed as design level considerations.

Assessment of ROSS/ICBS application-level technical options requires a review of the operational functions supported by each application and the application-level code and data structure-level tables supporting each operational function. As would be expected, the application level is highly dependent upon the data structure level of the overall system design. However, the two are not directly coupled:

- Addition of table columns to the ROSS and/or ICBS data structures could be performed without any significant impact to application-level code.
- Application-level changes do not necessarily require GUI-level modifications. However, GUI modifications are required if operational function is affected.
- Integration of the ROSS and ICBS data structures would require substantial application-level change to both ROSS and ICBS.

5.1 Location for "Application-Level" Changes

When discussing application-level technical options, it is important to note that the nature of the required application-level modifications depends on the development environment associated with the respective system (ROSS or ICBS), and the nature of the change that must be accommodated.

ICBS is an Oracle-based system developed using Oracle Forms. Several types of application-level modifications to the ICBS application are possible including:

- Changes to the Oracle Forms client-side GUI layout
- Changes to client-side triggers, PL*SQL procedures and functions
- Changes to stored procedures and Oracle control mechanisms embedded in server-side RDBMS.

ROSS has been developed in a Versata Vision JADE development environment with an Oracle 8.1.6 RDBMS. Several types of application-level modifications to the ROSS application are also possible including:

- Changes to the Java-based client GUI application
- Changes to Versata Logic Server (VLS) imbedded business rules
- Changes to server-side Java code
- Changes to stored procedures and Oracle control mechanisms embedded the Oracle 8.1.6 data structures

5.2 Application/Data Structure Alternatives

As noted above, the application-level code is highly dependent upon the data structure level of the overall system design. However, the two are not directly coupled. Below is a description of application-level

impact that would result from the use of each data structure-level technical option currently under consideration.

5.2.1 Data Translation

As indicated in Table 5-1 below, a data structure-level approach, which relies upon the addition of columns to existing ROSS/ICBS tables and the use of translation logic to map information between the ROSS and ICBS systems, results in the least impact to existing ICBS and ROSS applications. Under this option the ROSS and ICBS would exchange information, providing an increase in operational capability, with minimal changes to existing ROSS and/or ICBS operational functions. Also, minimal changes to the respective ROSS or ICBS GUI interfaces would be made.

5.2.2 Partial Integration

As indicated in Table 5-1 below, partial integration results in greater impact to the ROSS/ICBS applications, although not necessarily at the GUI level. So long as no change in operational capability is initiated, the application-level modifications required by a partial integration approach could be made in a manner transparent to the ROSS/ICBS user community.

5.2.3 Full Integration

As indicated in Table 5-1 below, full integration results in the greatest impact to the ROSS/ICBS applications. Such change would probably necessitate significant change to the respective ROSS/ICBS application, including the application GUIs, in order to take full operational advantage of the characteristics of the integrated data structure in the form of increased functional capability.

Table 5-1 Impact of Integration Alternative on ROSS/ICBS Application Levels

	ICBS GUI	ICBS Client Logic	ICBS RDBMS Logic	ROSS Client	ROSS VLS Logic	ROSS Server Java	ROSS RDBMS Logic	Translate Logic
Add Columns,X-late tables	low	low	low-med	low	low	low	low-med	high
Partial Integration	low-med	low-med	med-high	low-med	low-med	low-med	med-high	low-med
Full Integration	med-high	high	high	med-high	high	med-high	high	none

5.3 Assessment of ICBS Functionality

Table 5-2, below, defines the levels ROSS/ICBS operational capability, originally presented in Section 1.3, Background/Scope, in terms of ICBS capabilities, as defined in the ICBS System Design document. The number of screens and fields associated with each function have been summed in order to obtain quantitative values to assess the complexity of these capabilities.

These tables can be used in several ways: To validate that Statement of Work (SOW) requirements have been properly interpreted in terms of operational capability and scope; As a basis for “trimming” unnecessary operational capability, to reduce costs associated with any particular ROSS/ICBS alternative.

Table 5-2 Summary of Requested Capability in Terms of ICBS Functions

Design Document Section	Process Order/Request	Design Doc Sect# - Page#	Function Description	# Screens	# Fields	ICBS Tables
	Read NFES Catalog Information from ICBS					
4.3.2	NFES Catalog Information Report	4-40	Report which provides description, quantity available, and price information for a range of one or more ICBS cache item numbers as would appear in the NFES catalog	2	20	ICBS_Cache; ICBS_Cache_Item; ICBS_Cache_Item_Repos; ICBS_Hazmat; ICBS_Inv_Qty; ICBS_Item_Comp; ICBS_Item_Supp; ICBS_Kit_Pending_Ret_Item; ICBS_Supplier; ICBS_Cache_Item_Inv; ICBS_Acct_Prop;
			TOTALS Function Point Counts	**2**	**20**	**11**
	Read Cache Inventory Information from ICBS					
4.6.11	Stock Status (by cache ID; by Item Number)	4-102	Reports the current stock status of items in the cache.	3	33	ICBS_Cache; ICBS_Cache_Item; ICBS_Cache_Item_Repos; ICBS_Hazmat; ICBS_Inv_Qty; ICBS_Item_Comp; ICBS_Item_Supp; ICBS_Kit_Pending_Ret_Item; ICBS_Supplier; ICBS_Cache_Item_Inv;
4.2.2	Cache Item Component Availability Report (by cache ID; by all cache ID's; based upon requested quantity)	4-28	This report gives a summary of how many cache items (consolidated) can be built from the existing inventory levels. It also provides a summary of how many cache items (component) are short. This is a shortage kit and kit build summary report.	2	15	
4.2.1	Cache Item Component Report	4-26	This report generates a listing of all components required to create a cache item (consolidated).	2	15	
4.2.3	Cache Item Packing List	4-34	This report will generate the packing list that contains a list of all cache item (components) contained within the cache item (consolidated). This is a kit package list report.	2	25	
4.2.4	Kit Item Summary Report	4-36	This report will produce a list of all cache items (consolidated) that contain a particular cache item (component). Ex. A list of all kits that contain a shovel.	2	15	
4.3.1	Cache Item Summary	4-38	This report will print out all information pertaining to a specific cache item. It includes catalog, supplier, manufacturer and model information.	2	20	
			TOTALS Function Point Counts	**13**	**123**	**10**
	Designate items as critical within the NFES Catalog Data within ROSS					
4.4.2	Flag Critical Items	4-46	This is an input/edit screen used to flag cache item numbers that are critical. Upon entering this screen all critical items will be displayed sorted by cache item number. Deletions, insertions, and updates may be done for any item.	1	3	ICBS_Cache; ICBS_Cache_Item; ICBS_Hazmat; ICBS_Inv_Qty;
4.4.3	Critical Items Report For NICC	4-47	Generat critical item report that can be produced locally which will list all items flagged as national or local critical items.	1	9	
			TOTALS Function Point Counts	**2**	**12**	**4**
	Transmit and receive supply order/fill information between ROSS and ICBS					
			See Table 5-3, ICBS Order / Fill Related Function Points			
			TOTALS Function Point Counts	**40**	**523**	**33**
	Implement all interfaces through re-host / re-engineering of ICBS					
			See Table 5-4, All ICBS Related Function Points			
			TOTALS Function Point Counts	**190**	**2271**	**47**

Table 5-3 ICBS Order / Fill Related Functions

Design Document Section	Process Order/Request	Design Doc Sect# - Page#	Order (o) or Fill (f) Related	Function Description	# Screens	# Fields	ICBS Tables
1.1.2A	Define the Customer	1-2*	o	Define a new customer or update an existing one	1	8	ICBS_Acct_Code_Chg_Ref; *ICBS_Acct_Prop; *ICBS_BO_Item; *ICBS_Cache; *ICBS_Cache_Item; ICBS_Cache_Item_Chg_Ref; *ICBS_Cache_Item_Inv; ICBS_Cache_Item_Red; *ICBS_Cache_Item_Repos; *ICBS_Customer; ICBS_Hazmat; ICBS_Inv_Loc ICBS_Inv_Qty; *ICBS_Issue; *ICBS_Issue_Item; *ICBS_Item_Comp; *ICBS_Item_Supp; *ICBS_Kit_Instr; ICBS_Lov; *ICBS_Ord; ICBS_Ord_Num_Chg_Ref; *ICBS_Other_Ord_Acct_Code; *ICBS_Req_For_Supp; *ICBS_Req_For_Supp_Item; ICBS_Res_Req; ICBS_Return; *ICBS_Shipping; ICBS_Trans_Adjust; ICBS_Trans_Issue; ICBS_Trans_Issue_Tot; ICBS_Trans_Prop; ICBS_Trans_Rect; ICBS_Trans_Ret; ICBS_Trans_Seq_Ref; ICBS_Trans_Transform; ICBS_Transfer; ICBS_Users; * Denotes Fundamental Order/Fill Capability
1.1.2B	Define Incident Order	1-4*	o	Define a new incident order or update an existing one	1	14	
1.1.1/1.1.4.1	Input Incident Order and Produce Issue	1-6*	o/f	Input an order and generate issue, which includes the following steps: - Enter Order Header Information - Enter Requested Items - Issuing Kits with Accountable Property - Issuing Accountable Property	See Below		
	Enter Order Header Information	1-6*	o	Enter incident order header information	1	25	
	Enter Requested Items	1-8*	o	Enter incident order requested items	1	12	
	Issuing Kits with Accountable Property	1-11*	f	Issue kits with accountable property based on incident order requested items	1	8	
	Issuing Accountable Property	1-13*	f	Issue accountable property based on incident order requested items	1	8	
1.1.3/1.1.4.1	Retrieve Incident Order and Produce Issue	1-20*	f	To retrieve an order, enter the Request Group Number, Incident Order Number, or select it from the Pending Orders List. No issues have been produced for these orders.	1	25	
	Pending Orders List	1-22*	o/f	Generate list of pending orders for selection	1	7	
	Enter Issue Quantities	1-24*	f	Fill selected order with a specific quantity	1	13	
1.1.4.1	Other Orders	1-26*	o/f	Support entry and fill of orders not associated with incidents	2	37	
	Splitting charges across multiple account codes	1-30	f	Incident orders may only have a single charge code. Non-incident orders may be charged across multiple account codes	1	15	
1.1.4.1	Query Issues	1-32*	o	Allow issues to be retrieved by: “Order Number”, “Issue Number”, or both. After the query is made Issue records may be displayed and edited.	2	37	
1.1.4.1	Transferring Cache Items out of the Cache	1-36	f	Allow cache items to be transferred out of the local cache and issued to a sub-cache or staff. (i.e., items leave the cache but remain in the cache database)	3	44	
1.1.4.2	Correction	1-42*	o	Allow order information to be corrected, including: - header information - requested items - issued items	2	37	
	Select from List of Issues	1-46*	f	Allow issues to be searched, selected, and corrected, even if the issue number is not known.	1	6	
1.1.5.5	Update Issue Transportation Information	1-48	f	Allow issue transportation information to be entered and updated. Individual issued items may be cancelled and not carried.	2	30	
	Shipping Status Information Document	1-52	f	Allow shipping information to be printed, including items sent as well as cancelled items that were not sent.	1	16	
1.2.0	Canceling Orders and Issues	1-54*	o/f	Orders are in two states once the cache has received them. They are shipped or not. Non-shipped orders may be in one of many sub-states. An order may have been received by the cache and not entered into the computer system, an order may be entered in the computer system and the floor personnel are in the process of picking it, an order may be prepared for shipment and waiting to be loaded, and an order may have already shipped from the cache. Orders that fall into the first three categories may be canceled, but orders that have been shipped are to be processed as a return if the customer cancels any or all of it.	3	50	
1.3.1	Process Issue Shortage Notice	1-62	f	Allow adjustments to be made to an issue for items missing from a delivered shipment. This process is for non-accountable property items only. It is also not used for kits that were packed improperly. Ex. 10 shovels were sent to a fire but only 8 of them showed up. Two shovels would be credited to the issue then 2 shovels would be adjusted off the inventory as a loss.	1	12	
1.4.1	Reserve Cache Items	1-64	f	Allow cache items to be placed in 'RESERVE' status in anticipation of future need. This process will decrement the Qty RFI and increment the Qty Reserved by the quantity in 'Qty Reserved'. Any cache item, including kits and accountable property, may be reserved. Once items are placed in reserve status they must be issued or deleted from this process to remove them. Items will be grouped under the Customer Unit ID.	1	8	
	Reserved Item Listing	1-66	f	Allow reserved items to be queried and listed.	2	19	
1.5.1	Backorders	1-68	f	Provide a means to query by “order number” or “Cache Item Number” and view backorders to determine whether they should be filled or killed. The process will allow a back order to be selected and items issued to it or allow the order removed.	3	32	
	Backorder Listing	1-73	o/f	Allow a listing/printout of back order items to be generated.	3	24	
1.6.1	Transfer Cache Items	1-76	f	Allow items to be transferred from one project/incident to another. Issues may be transferred based upon “incident order type” or “other order type”	3	24	
1.6.2	Transfer Correction	1-84	f	Transfer Issues may be queried, selected for editing and corrected	1	12	
				TOTALS Function Point Counts	**40**	**523**	**37**

Table 5-4 All ICBS Related Functions

Design Document Section	Process Order/Request	Design Doc Sect# - Page#	Function Description	# Screens	# Fields	ICBS Tables
1.0	**Process Order/Request**	1-1	See Table 5-3, above.			
			TOTAL Function Point Count – Section #1	40	523	
2.0	**Process Incoming Items**	2-1	See ICBS Design Document for Descriptions			
2.1.1/2.1.3	Identify Account and Inspect Items	2-2		5	58	
	Enter Returned Item Status/Quantity	2-6		1	15	
	Retrieve Incident	2-9		1	4	
	Define Incident	2-10		1	12	
	Returning Accountable Property	2-11		1	8	
	Returning Kits	2-12		1	8	
	Enter Returned Items from Kit Contents	2-14		1	8	
	Enter Return Comments	2-15		1	2	
2.1.2/2.2.2	Workorder	2-16		1	13	
	Enter Workorder Information	2-16		1	5	
	Print Open Workorder	2-19		2	20	
	Close Workorder	2-21		2	40	
	Workorder Return Document	2-24		1	15	
2.1.4	Return Correction	2-26				
	Correct Returned Item Status/Quantity	2-26		1	18	
	Select from List of Returns	2-29		1	5	
	Correct Workorder Kits	2-30		1	9	
2.2.4	Process Receipts from Suppliers	2-32				
	Receipt by Purchase Document	2-32		1	5	
	Receipt by Purchase Order/Requisition	2-34		1	14	
	Receipt by Cache Item	2-36		1	13	
	RFI Items not on Purchase Document	2-38		1	7	
			TOTAL Function Point Count – Section #2	**26**	**279**	
3.0	**Warehouse Activities**	3-1				
3.1.1/3.1.2	Enter Item Information	3-2				
	Item Catalog	3-2		1	30	
	Item Inventory	3-7		1	7	
	Accountable Inventory	3-9		1	22	
	Hazardous Information	3-12		1	17	
	Kit Contents	3-14		1	6	
3.1.3	Request Cache Items	3-16		1	16	
3.1.4/3.1.5	Buildup/Breakdown Kits	3-20				
	Build Cache Kits/Items	3-21		2	16	
	Breakdown Cache Kits/Items	3-23		2	16	
	Build Kits Short or with Substitutions	3-25		1	13	
3.1.6	Enter Purchase Order Information	3-28		1	13	
	Enter Items on Purchase Order by Item Number	3-29		1	20	
	Enter Items on Purchase Order by Requisition Number	3-30		1	19	
3.1.7	Enter Item Supplier Information	3-32		1	17	
3.2.1/3.2.5	Determine Appropriate Disposal	3-34		1	15	
3.2.2	Reconcile Missing Cache Items	3-36		1	12	
3.2.3	Change Cache Item Storage Location	3-38		1	8	
	Enter Accountable Property to Move	3-40		1	9	
	Enter Primary Accountable Property Number of Kit to Move	3-42		1	6	
3.3.1/3.3.2	Archive Records	3-46				
3.3.3	Change Cache Item	3-47		1	3	

ICBS_Acct_Code_Chg_Ref;
*ICBS_Acct_Prop;
*ICBS_BO_Item;
*ICBS_Cache;
*ICBS_Cache_Item;
ICBS_Cache_Item_Chg_Ref;
*ICBS_Cache_Item_Inv;
ICBS_Cache_Item_Red;
*ICBS_Cache_Item_Repos;
*ICBS_Customer;
ICBS_Hazmat;
ICBS_Inv_Loc
ICBS_Inv_Qty;
*ICBS_Issue;
*ICBS_Issue_Item;
*ICBS_Item_Comp;
*ICBS_Item_Supp;
*ICBS_Kit_Instr;
ICBS_Lov;
*ICBS_Ord;
ICBS_Ord_Num_Chg_Ref;
*ICBS_Other_Ord_Acct_Code;
*ICBS_Req_For_Supp;
*ICBS_Req_For_Supp_Item;
ICBS_Res_Req;
ICBS_Return;
*ICBS_Shipping;
ICBS_Trans_Adjust;
ICBS_Trans_Issue;
ICBS_Trans_Issue_Tot;
ICBS_Trans_Prop;
ICBS_Trans_Rect;
ICBS_Trans_Ret;
ICBS_Trans_Seq_Ref;
ICBS_Trans_Transform;
ICBS_Transfer;
ICBS_Users;

Tables Not Used by Section #1 Functions:
ICBS_Kit_Pending_Ret_Item;
ICBS_Kit_WO_Item;
ICBS_Pending_Prop;
ICBS_Pending_Ret;
ICBS_Pending_Ret_Item;
ICBS_PO_Item;
ICBS_Purchase_Ord;
ICBS_Supplier;
ICBS_WO_Item;
ICBS_Workorder;

* Denotes Fundamental Order/Fill

Design Document Section	Process Order/Request	Design Doc Sect# - Page#	Function Description	# Screens	# Fields	ICBS Tables
3.3.4	Change Accounting Code	3-48		1	8	Capability
3.3.5	Change Order Number	3-50		1	7	
3.3.8	Restore Records from Archive	3-52				
	Restore Customer Records	3-53		1	2	
	Restore Transaction Records	3-53		1	7	
	Restore Order Records	3-53		1	3	
	Restore Item Catalog Records	3-54		1	1	
	Restore Inventory Records	3-54		1	1	
3.4.1	Enter Inventory Results	3-56				
	Enter Non Accountable Inventory Results	3-56		1	5	
3.4.2	Enter Local Item Costs	3-58		1	5	
3.5	Year End Processing	3-60		1	9	
			TOTAL Function Point Count – Section #3	**31**	**313**	
4.0	**Generate Reports/Output**	4-1				
	BATCH PRINTING	4-2		1	7	
4.1.1	Transaction Listing	4-4		3	26	
4.1.2	Account Transaction Report	4-8		1	7	
	Transactions by Accounting Code & Transaction Type	4-10		1	15	
	Transactions by Accounting Code & Order Number	4-11		1	15	
	Transactions by Order Number & Transaction Type	4-12		1	15	
	Transactions by Order Number & Document Number	4-13		1	15	
4.1.3	Billing Information Report	4-14		3	60	
4.1.4	Open Order Billing Report	4-20		2	25	
4.1.5	Incident Summary	4-22		3	32	
4.2.1	Cache Item Component Report	4-26		2	14	
4.2.2	Cache Item Component Availability Report	4-28		1	3	
	Components by Cache ID.	4-30		1	12	
	Components in all Cache ID's	4-31		1	12	
	Components for Requested Quantity	4-32		1	18	
4.2.3	Cache Item Packing List	4-34		2	24	
4.2.4	Kit Item Summary Report	4-36		2	13	
4.3.1	Cache Item Summary	4-38		2	22	
4.3.2	NFES Catalog Information	4-40		2	26	
4.3.3	NFES Box Label	4-44		2	10	
4.4.2	Flag Critical Items	4-46		1	4	
4.4.3	Critical Items Report For NICC	4-47		1	8	
4.4.4	Local Critical Item Report	4-48		2	11	
4.5.1	Address Code Listing	4-50		2	12	
4.5.2	Incident/Project List	4-52		2	17	
4.6.1	Accountable Property Report	4-54		2	20	
	Accountable Property Status	4-58		2	22	
4.6.2	Accountable Property Disposal	4-62		2	25	
	Non Accountable Property Disposal	4-66		2	21	
4.6.3	Cache Item Below Minimum Report	4-70		2	24	
J 4.6.4	Cache Item Above Maximum	4-74		2	24	
4.6.6	Duplicate Location Report	4-78		2	11	
4.6.7	Inventory Worksheet	4-80		1	5	
	Non Accountable Inventory Worksheet	4-82		1	11	
	Accountable Property Inventory Worksheet	4-84		1	20	
4.6.8	Issue	4-88		3	80	
4.6.9	Redistribution	4-92		2	17	
4.6.10	Return Documents	4-94		1	3	
	Return	4-94		1	24	
	Close Workorder	4-98		1	26	
	Workorder Return Document	4-101		1	30	
4.6.11	Stock Status	4-102		1	7	
	Stock Status by Cache ID	4-104		1	18	
	Stock Status by Item Number	4-105		1	18	
4.6.12	List of Items with Expiration	4-106		1	15	
4.6.13	Inventory Discrepancy	4-108				

Design Document Section	Process Order/Request	Design Doc Sect# - Page#	Function Description	# Screens	# Fields	ICBS Tables
	Enter Non Accountable Inventory Results	4-108		1	5	
	Non Accountable Inventory Discrepancy Report	4-110		1	18	
4.6.14	Cache Item Transaction Listing	4-114		1	25	
4.6.15	Stock History Report	4-118		2	38	
4.6.16	Alaska Stock Display	4-122				
	Stock Status Alphabetical Display	4-122		1	2	
	Stock Status Informational Display	4-124		2	40	
	Stock Status History Display	4-126		1	10	
4.7.1	Fire Loss/Use	4-130		2	25	
4.8.1	Open Purchase Order Listing	4-134		2	23	
4.8.2	Open Requisition Listing	4-136		2	20	
4.8.1.6	Item Supplier List	4-138		2	34	
4.8.4	Item Purchases Report	4-140		2	20	
4.9.1	Prepare Workorder	4-144				
	Enter Workorder Information	4-144		1	13	
	Split Workorder Line	4-146		1	5	
4.9.2	Print Open Workorder	4-147		1	34	
			TOTAL Function Point Count – Section #4	**93**	**1156**	
			TOTALS Sections 1-4	**190**	**2271**	**47**

5.4 Application-Level Technical Option Considerations

5.4.1 Considerations for Partial Integration and/or Full Integration

Based upon an evaluation of Table 5-2 through Table 5-4, if a partial data integration approach is adopted, then it may make the most sense to integrate data structures supporting the critical order/fill process first (See Table 5-3, items indicated with *'s), leaving as a separate application the processing of incoming items, warehouse activities, and their related reports (Table 5-4). In this manner, maximum operational capability can be provided at reduced initial cost.

However, due to the relational nature of the ICBS database, significant consideration would be required to assure compatibility of the redesigned portion of the ICBS data structures with the original ICBS. Needless to say, this approach represents a significant development undertaking. Such changes would necessitate significant application-level redesigns with respect to both ROSS and ICBS. Full integration would necessitate even greater change to both applications

5.4.2 Considerations for Data Translation

Issues associated with both partial and full integration can be entirely avoided by developing a database/application layer that supports the mapping of ROSS/ICBS data between the two structures. The advantages of this approach are:

- The original ICBS and ROSS applications would be unaffected
- The data structure modifications required to support the translation process would be minimal
- Not all translation routines would need to be implemented at once.

As previously stated, it may make the most sense, to integrate data structures supporting the critical order/fill process first (See Table 5-3, items indicated with *'s), leaving as a separate application the processing of incoming items, warehouse activities, and their related reports (Table 5-4).

The disadvantage of the "Data Translation" approach is that, if all ICBS information were to be translated, the translation effort itself would be a significant piece of development. Furthermore, other than the exchange of information between ROSS and ICBS, no added functionality would be provided.

5.4.3 Hybrid Approach

One way to obtain immediate benefits while providing a modular growth path to a more highly integrated ROSS/ICBS environment is to use a hybrid of both data translation and data structure integration techniques. Under this approach data translation modules would be incrementally developed to translate data in support of prioritized ROSS and ICBS functions. These translation modules would be developed at the rate needed and at the rate that the sponsoring organization can afford. As part of the translation process, temporary tables would be implemented which would be, in fact, a small component of an overall ROSS/ICBS integrated database design. As revisions to the respective ROSS or ICBS modules are performed, each revision would build to that portion of the integrated data structure that is mature, negating the need for the translation modules with respect to that particular portion of the overall structure. In this manner, immediate needs can be achieved with minimal resources expended; however, implementation of long-term integration planning can be achieved

6. Hardware Infrastructure Technical Options

This section addresses the hardware infrastructure requirements associated with each type of Oracle row-level replication. It also identifies the operational advantages and capabilities associated with each requirement.

In both read-only and updateable snapshot environments, remote databases can be disconnected from master sites without having any impact on the replication environment. The remote site(s) can be reconnected to the master site and refreshes will be performed, thereby refreshing the associated snapshot tables. In addition, in the case of updateable snapshots, DML directed back to the master site will be sent, and assuming no conflicts arise, applied to the master site database.

In either of these two types of replication environments, there are no hardware infrastructure requirements specific to replication. By choosing either of these types of replication, the assumption is made that the potential risk of having remote database nodes going down is acceptable. Of course, each DBA for each of the participating databases should be following his/her own backup and disaster recovery plans.

In the case of advanced replication, however, if synchronous propagation is being used (i.e., the two-phase commit protocol is being utilized), each participating database server must not go down. A failure of one of the database servers in this environment will bring the entire system down. All DML activity will cease, and the applications associated with these databases will hang. Additionally, recovery will involve not only the database server that went down but also all participating databases in the environment. Therefore, it is critical to provide as much redundancy as possible in terms of disk mirroring and network connectivity.

6.1 Primary Platform Considerations

The ROSS Vision-JADE/ Oracle internal architecture provides many viable options with respect to the physical platform(s) that support the operational application. The Versata Logic Server (VLS) provides the ability to add VLSs transparently, as processing demands require, and allows data to be distributed transparently across multiple Oracle RDBMSs. As a result, the number, processing capacity, and storage requirements of the initial ROSS hardware platforms can be initially calculated based upon existing known requirements. Later, as functional capabilities, user populations, and the processing/storage requirement increases, the ROSS platform infrastructure can be modularly expanded as necessary to meet these new requirements.

The ICBS is a pure Oracle-based environment. As a result there are fewer possibilities for expansion. Oracle RDBMSs can be pooled but only with a substantial increase in overhead, increased maintenance, and decreased performance per node. As described above, Oracle advance replication alternatives cannot easily recover from the loss of a single node. Therefore, if a multi-master, real-time replication alternative is chosen for ROSS/ICBS data sharing, there will be requirements for greater hardware-level reliability and, hence, significant hardware-level ramifications. Reliability and redundancy considerations and the mechanisms available to address them are presented in Table 6-1, below.

Furthermore, ROSS/ICBS data-sharing alternatives that call for consolidation or integration will increase the RDBMS storage requirements for the consolidation node by roughly a factor of 13 (i.e., the combined storage requirements of all 13 ICBS repositories).

Table 6-1, ROSS / ICBS Platform Considerations

IOC Platform Considerations	Related Platform Environment Options
Application Level Reliability	Redundant application server platforms Hot-standby application server software configurations Operating system/application RAID storage

IOC Platform Considerations	Related Platform Environment Options
Data Level Reliability	Redundant RDBMS server platforms Hot-standby RDBMS software configurations RDBMS level mirroring (implies processor/network overhead) RAID-5 based data storage RAID-5 hardware level mirroring (minimal overhead)
Increased Storage Requirements	Expandable internal RAID-5 storage Extremely expandable RAID storage towers
Increased Processing Requirements	Additional dedicated Vision VLS processors/platforms Separate RAID storage towers configured for RAID-5 hardware level database mirroring (Avoids processor loads associated the RDBMS level software mirroring)
Acceptable off-line recovery time	Hot-standby application and RDBMS configurations Redundant application and RDBMS platforms RAID-5 O/S and Database Storage

6.2 Platform Alternatives

The sections below present a viable platform configuration alternatives and provide a brief discussion of the capabilities and deficiencies of each alternative. An assumption made in each alternative is that the platform environment will be based upon RISC 6000 platform loaded with the IBM AIX operating system.

The typical server is assumed to be the equivalent of an RS/6000 Enterprise server model J50 with 2 GB memory, (6) 200 Mhz 2-way processors, 18.2 GB internal hard drive array, 12.6 external hot swappable arrays, and two 10/100 Mb Ethernet cards for connectivity to the local LAN. The model J50, configured as described above, was selected as the baseline based upon its proven high reliability, redundancy and compatibility with the existing ROSS deployment environment.

6.2.1 Minimum Configuration

Error! Reference source not found. presents a single CPU platform configuration that could, theoretically, serve either as the ROSS automated information system or as an ICBS RDBMS platform. However, serious hardware-level reliability issues are associated with such a configuration. Therefore, this option is not presented as a serious alternative for either ROSS or ICBS.

Capabilities:

- Internal RAID-5 facilitates O/S, application, and data recovery from the loss of any given single disk drive failure.
- Hot-swappable RAID-5 array minimizes downtime due to single drive failures.

Deficiencies:

- Failure of a single CPU and/or RAID drive results in loss of system service.
- Failure of more than one drive results in unrecoverable loss of data.
- Processor deficiencies may be experienced in servicing large number of simultaneous users sessions.

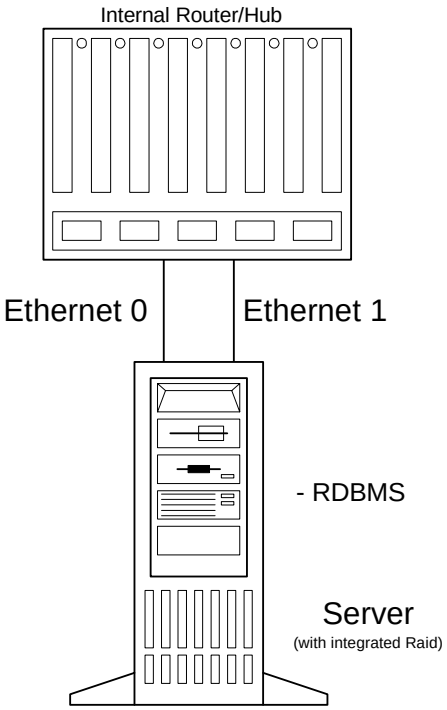


Figure 6-1 Minimum Configuration

6.2.2 Partially Redundant Configuration

Error! Reference source not found. presents a dual CPU platform configuration that provides an initial level of redundancy at the CPU and operational data storage level. Such redundancy provides a large increase in overall reliability and redundancy over the single CPU platform configuration.

Capabilities:

- All capabilities of single the platform configuration apply, with additional enhancements:
- Failure of a single CPU platform or drive results in degraded but no loss of service.
- Failure of two drives on any single platform does not result in loss of data or loss of service.

Deficiencies:

- Mirroring operational data via the RDBMS software level detracts from overall available processing capability.
- Resynchronization of a failed node at the data level is greatly complicated in multi-master replication configurations.
- Processor deficiencies may be experienced in servicing large number of simultaneous users sessions.

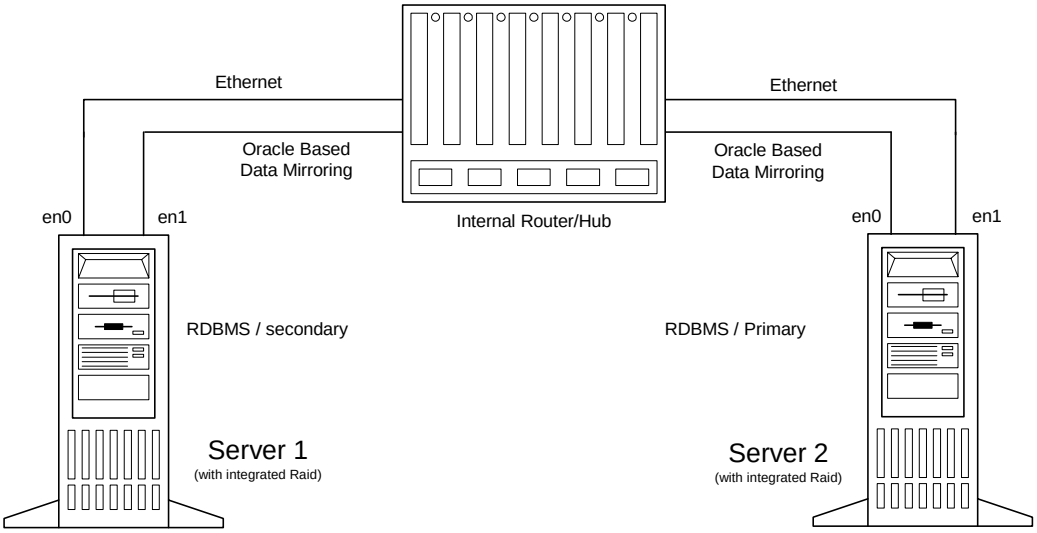


Figure 6-2 Partially Redundant Configuration

6.2.3 Fully Redundant (CPU Level and Data Level)

Error! Reference source not found. presents the same dual CPU platform configuration introduced in Section 6.2.2, with the addition of two SCSI RAID-5 storage towers. The RAID storage towers increase overall CPU processing power by allowing operational data to be mirrored between the two towers themselves at the hardware level via the SCSI bus. This enhancement avoids CPU processing requirements associated with operational data mirroring at the RDBMS level via the network. Furthermore, recovery and resynchronization of RDBMS nodes in a multi-master replication environment is greatly simplified. In addition, the RAID towers greatly increase overall operational data storage capacity (up to 256 GB).

Capabilities:

- All capabilities of the dual platform configuration apply, with the following enhancements:
  - Reduction of CPU processing requirements associated with operational data mirroring
  - Increased storage expansion capacity

Deficiencies:

- Processing restrictions may be experienced in handling large numbers of user sessions.

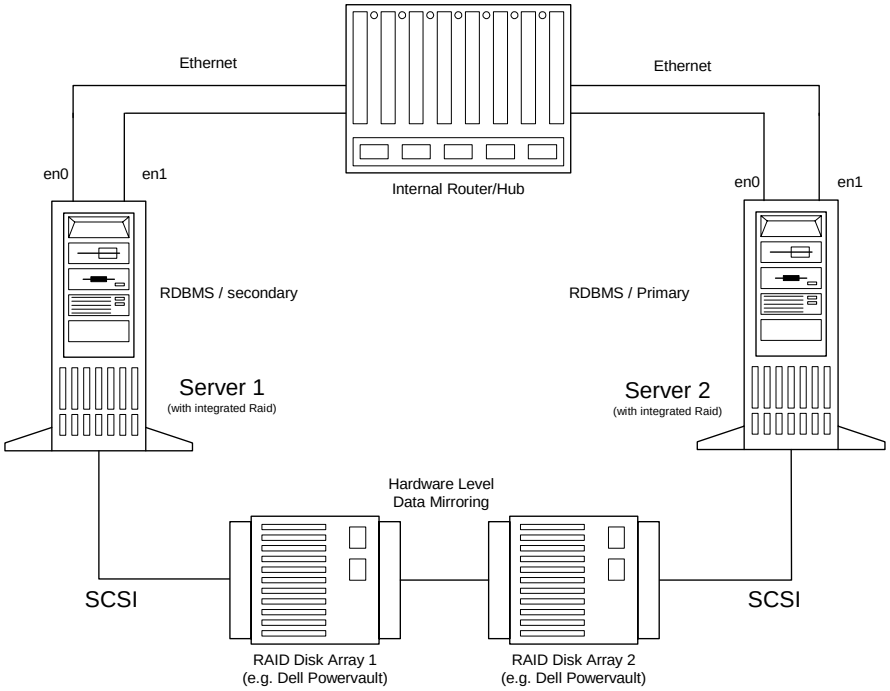


Figure 6-3 Fully Redundant Configuration (with Improved Data Storage)

6.2.4 Fully Redundant Modularly Expanded Configuration

Figure 6-4 presents the same dual CPU platform/SCSI RAID-5 tower configuration presented in Section 6.2.3, with the addition of one or more additional CPU platforms. These additional CPUs provide greater load-balanced CPU capacity where it is needed most, in support of Versata Logic Server (VLS) mid-tier processing. They also allow separate CPU platforms to be dedicated to support webserver and RDBMS functions.

Figure 6-4 represents the potential configuration and use of additional CPU platforms, as they are introduced within the infrastructure configuration. As functionality and user communities grow, the greatest increase in demand for CPU cycles VLS where additional functionality leads to increased code size/complexity and an increased user community leads to a linearly proportional increase in the number of simultaneous server-side processes supported by the VLS CPU. Fortunately, Vision JADE allows multiple VLS servers to be introduced seamlessly into the operationally deployed environment. Processor load is dynamically balanced between each VLS server to optimize the use of available CPU cycles across available processing platforms, freeing up a larger number of CPU and resources to support webserver and RDBMS functions regardless of the configuration.

Capabilities:

- All capabilities of the dual platform configuration (Section 6.2.2) apply.
- If separate RAID SCSI towers are included, all additional capabilities reflected in Section 6.2.3 apply.
- VLS load balancing provides optimal use of VLS CPU cycles and frees resources for web server and RDBMS functions respectively.

Deficiencies:

- None

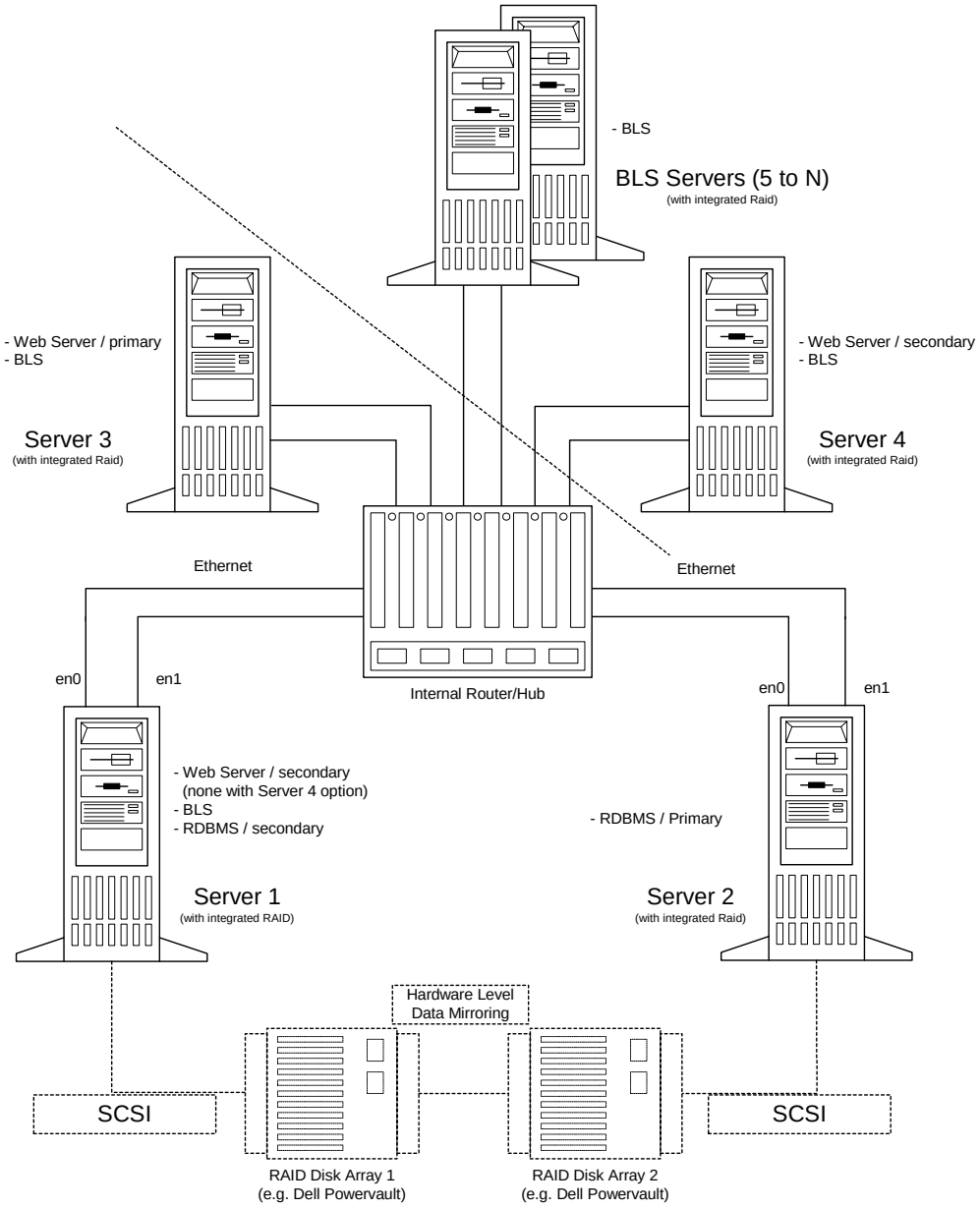


Figure 6-4 Fully Redundant Modularly Expanded Configuration

6.3 Hardware-Level Recommendations

Table 6-2 below presents a summary of recommended hardware-level options recommended for each ICBS cache node of the respective database infrastructure technical options presented in Section 3.

As indicated, the multi-master replication approach requires the higher level of hardware infrastructure reliability provided by a fully redundant configuration. This added reliability is required to avoid difficulties associated with re-synchronizing a failed multi-master node. The added cost of such redundancy would be required for each of the 13 ICBS nodes.

Oracle nodes in read-only and updateable configurations are easily resynchronized and therefore do not require such added redundancy. This approach avoids considerable cost and may allow deployment of read-only and updateable configurations without changes to the respective ICBS node infrastructures.

One or more of the proposed alternatives call for consolidation of the ICBS caches onto a single RDBMS. Should any one of these alternatives be adopted, the added hardware-level reliability of a fully redundant system is recommended. Furthermore, the consolidated external RAID-5 storage must be sized to store all 13 ICBS repositories, the ROSS database, expected growth projections, plus an additional 30% for RAID-5 overhead.

Table 6-2 ICBS Hardware Level Recommendations

RDBMS Infrastructure Option	HW Configuration (Non-Consolidated Options)	HW Configuration (Consolidated Options)
Updateable Snapshots	Partially Redundant Configuration - See Section 6.2.2	Fully Redundant - See Section 6.2.3
Multi-Master Replication	Fully Redundant - See Section 6.2.3	Fully Redundant - See Section 6.2.3

Table 6-3 below presents a summary of recommended hardware-level options recommended for the ROSS hardware architecture for each database infrastructure technical options presented in Section 3. Given the critical nature of the ROSS application to dispatch operations, the fully redundant configuration is recommended regardless of the RDBMS infrastructure option selected. Such a configuration is easily expanded to meet increased demands due either to increased ROSS workload or to future ROSS/ICBS integration.

Table 6-3 ROSS Hardware Level Recommendations

RDBMS Infrastructure Option	Recommended HW Configuration	Expanded HW Configuration
Updateable Snapshots	Fully Redundant - See Section 6.2.3	Modularly Expanded, Fully Redundant Configuration - See Section 6.2.4
Multi-Master Replication	Fully Redundant - See Section 6.2.3	Modularly Expanded, Fully Redundant Configuration - See Section 6.2.4

7. Network Infrastructure Technical Options

There are two measurements that are necessary for estimating the network infrastructure requirements of a replicated environment. These are data flows generated by transaction processing and data flows generated by replication activity. Transactional data flow can be approximated as:

- (bytes/transaction) X (peak transaction count/second)

Thus, in order to assess the network infrastructure requirements of the various alternatives proposed in this study, one must have not only a good understanding of the current level of transaction processing occurring in both the ROSS and ICBS environments, but also the probable future growth rate of this processing in both environments. Additionally, one must also have a reasonable estimate of the transactional data flow associated with the projected replication activity that will occur in this environment as well as the projected future growth pattern of this activity.

The inputs that would be required for assessing replication activity are:

- Total count of tables to be replicated in the environment
- Average number of records in each of these tables that will be updated per refresh and the average size of the record to be updated
- The periodicity of the refresh rate

It is advisable to multiply these estimates by a factor of 2 or 3. Operational behavior and load characteristics inevitably will change when the new replicated environment becomes operational.

In order to assess the network infrastructure requirements for each of the proposed alternatives, additional information will be needed on the current and projected transactional activity of the current ROSS and ICBS environments. Also, additional information will be needed in order to estimate the flow of transactional data associated with the projected replication environments proposed in the alternatives in this study.

8. Proposed Alternatives

The following sections provide a pictorial representation and synopsis of each ROSS/ICBS integration alternative considered by this study. Each alternative employs one or a combination of the types of Oracle row-level replication. The alternatives have been purposefully constructed so that each successive design would require increasing levels of modification at both the database and application level. Equally important, each successive alternative is an extension of the previous one, with enhanced capability provided by increased integration at the database infrastructure, database structure, and/or application level. Each successive alternative provides increased integration of the ROSS and ICBS data, data structures, and operations.

Also, the following alternatives present a fail-over/high-availability capability in the ICBS environment:

- Alternative #2a: Fail-over/high-availability is provided by establishing and maintaining identical data structures and data at all of the participating database servers in the multi-master replicated environment. If any one of the remote database servers should go down, the clients are directed automatically to an alternate database server.
- Alternatives #2b and #3a: Fail-over/high-availability is provided by the use of Oracle Parallel Server (OPS) in the ICBS environment. OPS is a technology that allows one physical database to be accessed by database instances running on two or more computers. This architecture affords high-availability because if one of the machines hosting an instance fails, processing can continue on the remaining node(s). Oracle8 has advanced the technology so that in some cases, users' connections are remapped to a different node transparently. In addition, in an OPS environment, each of the participating nodes can be used for processing.

In addition, Alternative #3a provides a fail-over/high-availability capability between the ROSS and ICBS environments through the creation of a multi-master replicated environment between the ROSS and consolidated ICBS database.

8.1 No Cost Alternative

8.1.1 Description/Rationale

The No Cost Alternative (Figure 8-1), included in the current ROSS Build Task, has the least impact from both the database and application standpoints. It involves using database links to create identical copies of the relevant tables containing the Catalog and Cache Inventory information from each individual ICBS database. The ROSS report application reads the data from the copied tables of the individual ICBS databases. Updates to the copied tables are periodic and require update to the entire set of tables, not just the rows that have changed since the last update. This approach does not include translation of the ICBS information into the current ROSS data structures.

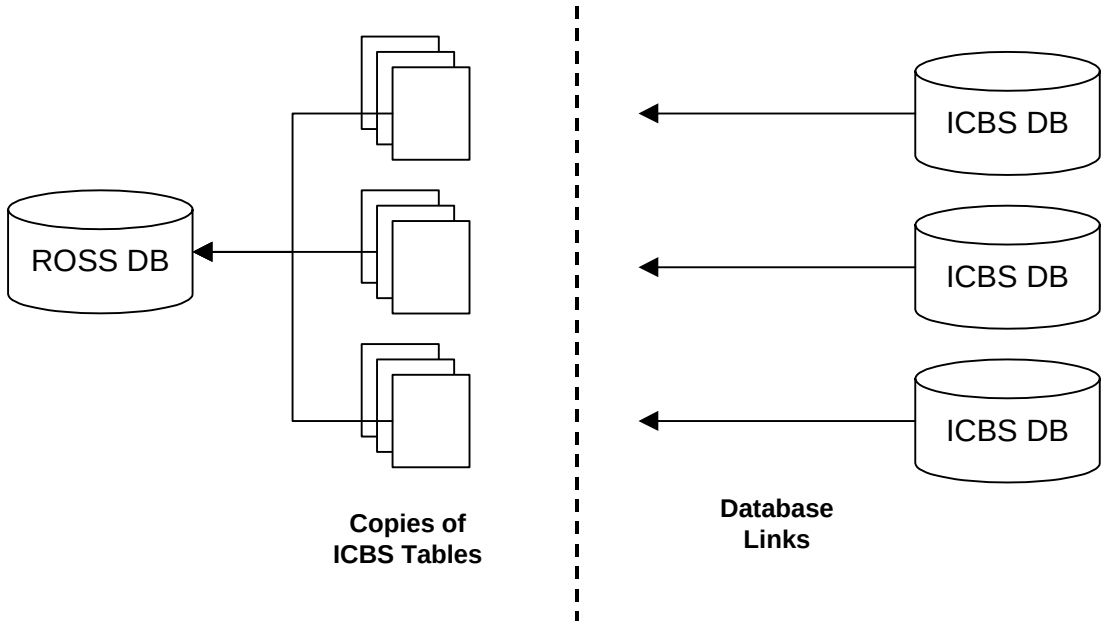


Figure 8-1 No Cost Alternative – Database Links

8.1.2 Database Impact

ICBS

- Read-only access to each of the ICBS databases with no alteration to the ICBS data structures.

ROSS

- Implementation of snapshot and merge tables within the ROSS database.

8.1.3 Application Impact

ICBS

- No impact

ROSS

- Addition of a report to view the ICBS Catalog and Cache Inventory Information via a ROSS business object. The business object supporting this view would most likely be a query object.

8.2 Alternative #1, Updateable Snapshots/Translation

8.2.1 Description/Rationale

In Alternative #1 (Figure 8-2) updateable snapshots are made of the relevant ICBS tables. A translation process occurs to provide read access to the ICBS data (i.e., going from right to left). In order to support write access in the ROSS application to the ICBS data, there is a second set of translation layers proceeding from the ROSS database to the ICBS databases (i.e., going from left to right). The first translation layer consists of stored triggers and procedures, which convert the structure of this data into the structure of the translation tables. The next translation layer refreshes the snapshot tables. The updated snapshot tables then update each master site ICBS database. Logical and physical data integration are accomplished within the ROSS environment by means of the translation layers and tables.

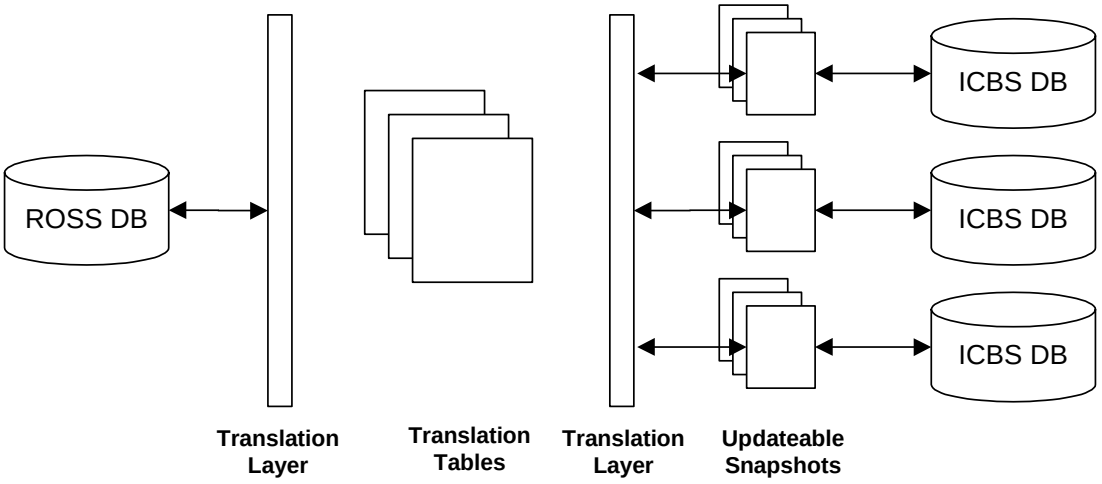


Figure 8-2 Alternative #1 - Updateable Snapshots

8.2.2 Database Impact

ICBS

- Addition of two columns in each individual ICBS database to serve as a concatenated key uniquely identifying records across all ICBS databases. The first column would be an alpha (a site identifier) and the second column would be populated by a sequence.
- Possible addition of a column providing a date/time stamp for an agreed upon time standard e.g. GMT.
- Setup and implementation to support updateable snapshots at the master site.

ROSS

- Addition of the same two columns added to the ICBS tables (as described above) to the corresponding tables containing Catalog and Cache Inventory information, serving as a

concatenated key to uniquely identify Catalog and Cache Inventory records across the ICBS and ROSS databases.

- Possible addition of a column providing a date/time stamp for an agreed upon time standard (e.g., GMT).
- Setup and implementation to support updateable snapshots at the snapshot site.

8.2.3 Application Impact

ICBS

- No impact

ROSS

- Modification of code at the application logic layer used corresponding to the changes in the database structure for the Catalog and Cache Inventory records.

8.3 Alternative #2a, ICBS Multi-Master/Updateable Snapshots

8.3.1 Description/Rationale

In Alternative #2a , physical data integration is accomplished in the ICBS environment by creating a multi-master replication environment using asynchronous propagation. Asynchronous propagation means that DML is collected in a deferred queue and then pushed to the participating databases at a user specified interval. The significant characteristic of the use of asynchronous propagation in a multi-master replicated environment is that if one of the participating remote database servers does go down, the multi-master environment is not affected. Logical data integration still occurs within the ROSS environment as in Alternative #1. This approach provides updateable snapshots allowing DML updates from the ROSS database to the ICBS environment.

This alternative will require that all participating databases will be identical both with respect to data and structure. This will require that each database server have adequate storage capability to handle the additional data associated with the other 12 remote ICBS cache sites.

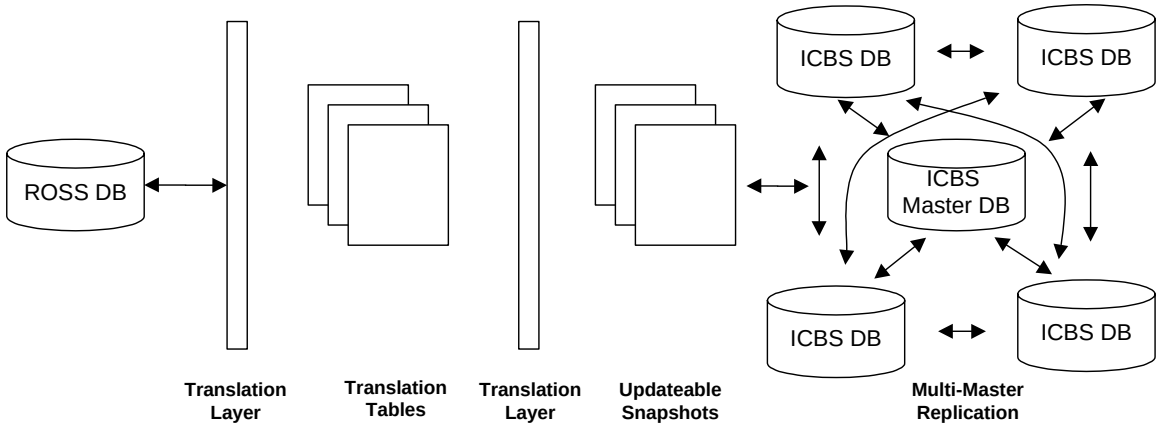


Figure 8-3 Alternative #2a - ICBS Multi-Master/Updateable Snapshots

8.3.2 Database Impact

ICBS

The impact would be identical to Alternative #1, except it would require:

- Set up and implementation to support a multi-master replication environment to involve all of the ICBS databases
- Setup and implementation of updateable snapshots at the master site
- Addition of a column providing a date/time stamp for an agreed upon time standard (e.g. GMT)

ROSS

- Identical to Alternative #1.

8.3.3 Application Impact

ICBS

- Identical to Alternative #1

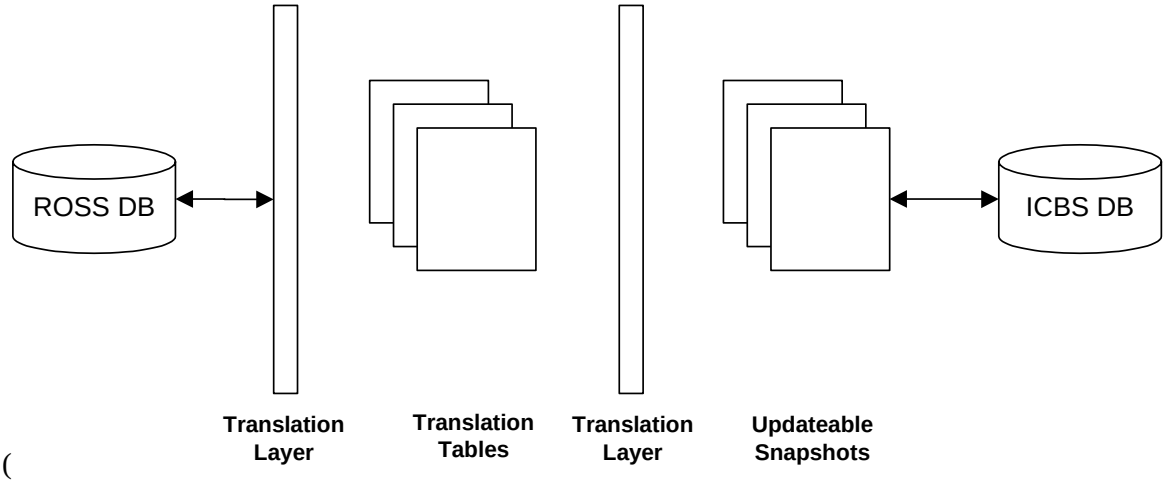
ROSS

- Identical to Alternative #1

8.4 Alternative #2b, ICBS Consolidation/Updateable Snapshots

8.4.1 Description/Rationale

Alternative #2b



(

Figure 8-4) achieves physical data integration through consolidation of the individual ICBS site databases within the ICBS environment. Physical data integration using this mechanism will involve changes to the ICBS application, since the ICBS individual site applications will no longer be interfacing with their own databases. Logical integration still occurs within the ROSS environment as in Alternative #1 and #2a. This alternative, like alternative #2a, uses updateable snapshots to allow writes to occur to the consolidated ICBS database.

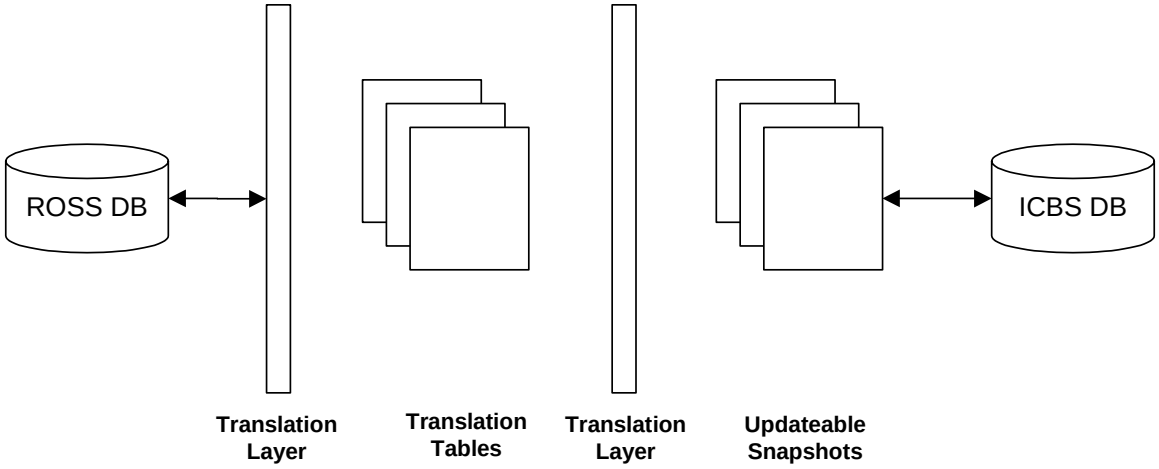


Figure 8-4 Alternative #2b - ICBS Consolidation/Updateable Snapshots

8.4.2 Database Impact

ICBS

- Addition of two columns in the consolidated ICBS database to serve as the new primary key uniquely identifying records across all legacy ICBS databases. The first column would be an alpha (a site identifier) and the second column would be populated by a sequence.

- Modification of any table relations, indexes, sequences, as well as any other database objects that would be impacted by these PK changes.
- Possible addition of a column providing a date/time stamp for an agreed upon time standard (e.g., GMT).
- Setup and implementation to support updateable snapshots at the master site.

ROSS

- Addition of the same two columns added to the ICBS tables (as described above) to the corresponding tables containing Catalog and Cache Inventory information, serving as a concatenated key to uniquely identify Catalog and Cache Inventory records across the ICBS and ROSS databases.
- Possible addition of a column providing a date/time stamp for an agreed upon time standard (e.g., GMT).
- Setup and implementation to support updateable snapshots at the snapshot site.

8.4.3 Application Impact

ICBS

- Modification of the application to use the new multi-column PKs used to uniquely identify catalog and cache inventory records as well as to accommodate any additional changes to the structure of any of the related tables and table relations affected by the primary key changes, as well as any other related database objects.

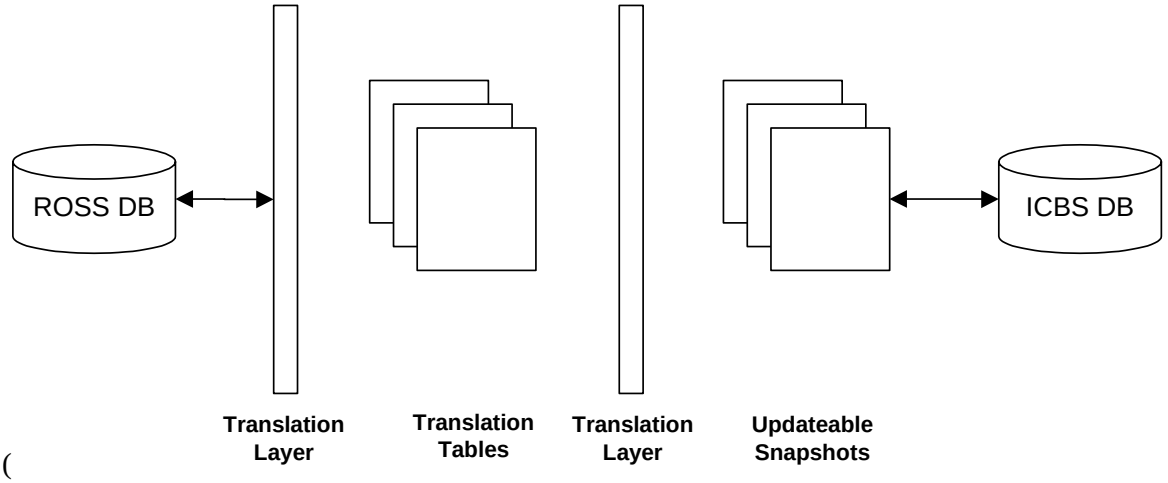
ROSS

- Modification of the application to use the new multi-column PKs used to uniquely identify catalog and cache inventory records as well as to accommodate any additional changes to the structure of any of the related tables and table relations affected by the primary key changes, as well as any other related database objects.

8.5 Alternative #3a, Identical Schemas/Multi-Master

8.5.1 Description/Rationale

Alternative #3a



(

Figure 8-4) achieves physical data integration through consolidation of the individual ICBS site databases within the ICBS environment as in Alternative 2b with one significant difference. The data structures of both ROSS and ICBS are integrated at the logical level. Physical data integration using this mechanism will involve changes to both the ROSS and ICBS applications, since the ICBS individual site applications will no longer be interfacing with their own databases and the logical data structures will more than likely have changed. This alternative maintains the individual databases of ROSS and ICBS. This alternative provides for the ability to replicate not only DML but also DDL. So, changes to database objects in one database i.e. the ICBS database can be automatically replicated to the ROSS database.

Figure 8-5 Alternative #3a - Identical Schemas/Multi-Master

8.5.2 Database Impact

ICBS

- Restructuring of the database at both the physical and logical levels
- Data migration from previous data structures to new data structures

ROSS

- Same as for ICBS
- Data migration from previous data structures to new data structures

8.5.3 Application Impact

ICBS

- Modification of the application due the changes in the logical and physical data structures of the database.

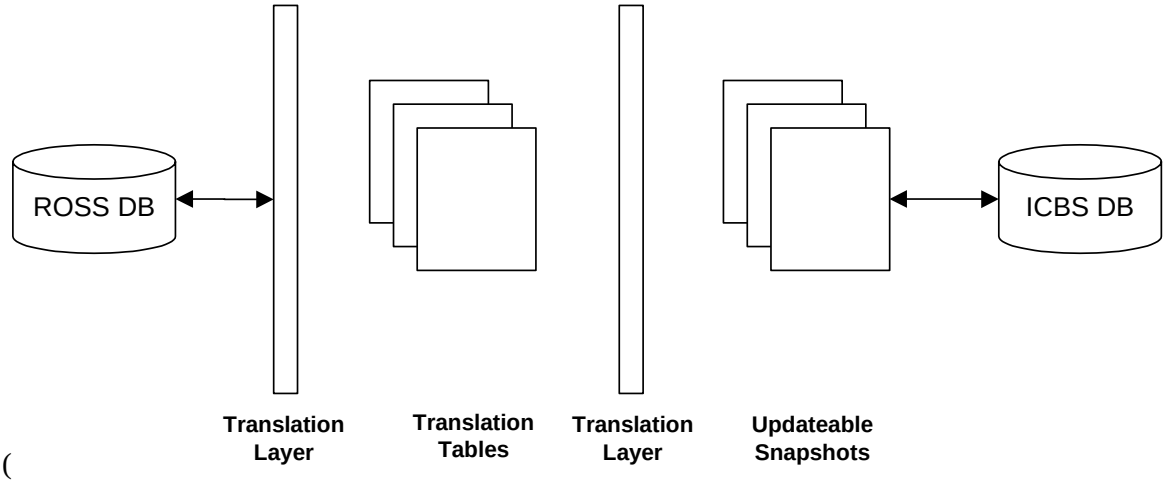
ROSS

- Same as for ICBS

8.6 Alternative #3b, Identical Schemas/Updateable Snapshots

8.6.1 Description/Rationale

Alternative #3b



(

Figure 8-4) is identical to Alternative #3a except that the interface between the two databases is achieved with updateable snapshots as in Alternative #2b.

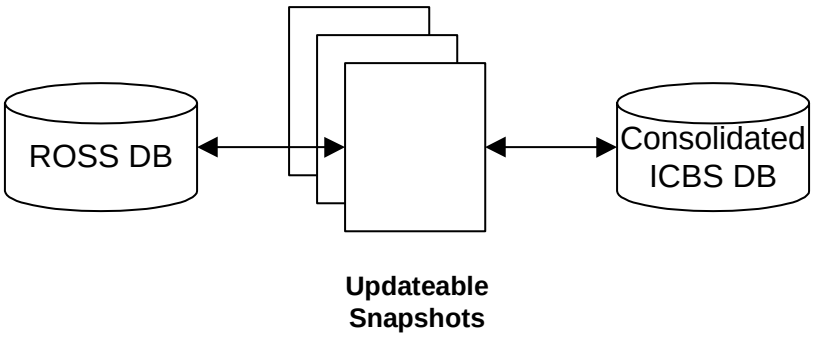


Figure 8-6 Alternative #3b - Identical Schemas/Updateable Snapshots

8.6.2 Database Impact

ICBS

- Same as Alternative #3a except that involves implementing updateable snapshots instead of multi-master replication.

ROSS

- Same as Alternative #3a except that involves implementing updateable snapshots instead of multi-master replication.

8.6.3 Application Impact

ICBS

- Same as Alternative #3a

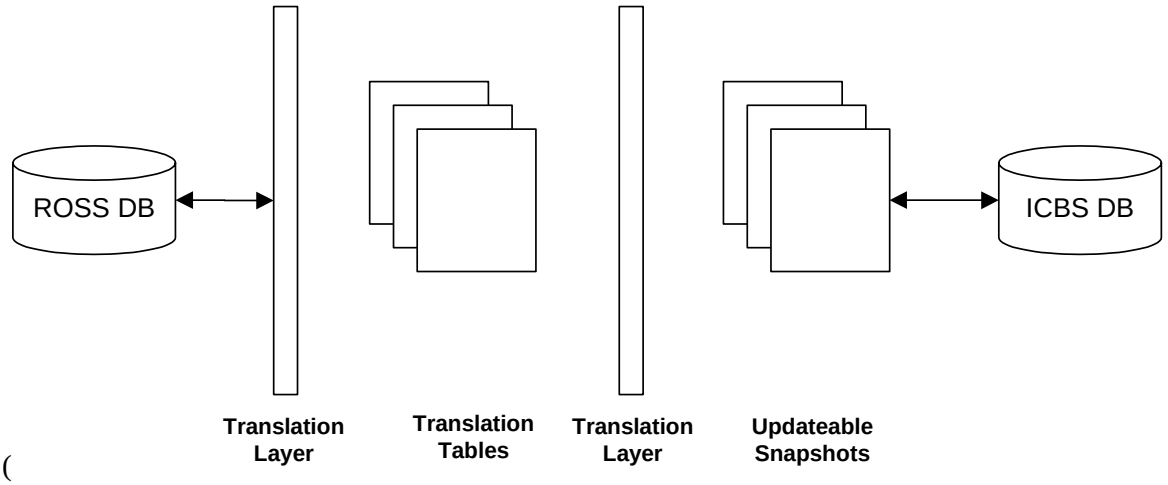
ROSS

- Same as Alternative #3a

8.7 Alternative #3c, Integrated ROSS/ICBS Database

8.7.1 Description/Rationale

Alternative #3c



(

Figure 8-4) is identical to Alternative #3a except that both ROSS and ICBS interface to the same database.

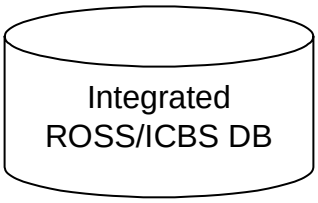


Figure 8-7 Alternative #3c - Integration ROSS/ICBS Database

8.7.2 Database Impact

ICBS

- Same as Alternative #3a

ROSS

- Same as Alternative #3a

8.7.3 Application Impact

ICBS

- Same as Alternative #3a

ROSS

- Same as Alternative #3a

9. Assessment of Advantages/Disadvantages

Table 9-1 identifies the advantages and disadvantages/risks associated with each alternative.

Table 9-1 ROSS/ICBS Integration Options - Disadvantages and Risks

Alternative	Advantages	Disadvantages/Risks
No Cost	Minimal impact at both the database and application levels Technically simple solution Quick implementation Remote database could be disconnected for extended periods without harm to the read only environment Administration of read only snapshot environment is relatively simple and straightforward	No integration at data level Data that is shared is neither real-time nor updateable
#1	Minimal impact at both the database and application levels Updateable snapshots provide read and write capability from the ROSS database back to the ICBS databases Remote database could be disconnected for extended periods without harm	The use of updateable snapshots requires the use of the Oracle advanced replication facilities. This adds an additional level of complexity to the environment both in terms of initial setup as well as general ongoing maintenance and administration Updateable snapshots will require an increased level of coordination amongst the participating database sites i.e. ROSS and ICBS databases Security: the initial setup and ongoing maintenance and administration of the security policy will be significantly greater in a replicated environment Transaction management: the DBA responsible for administering this environment must be careful that the number of transactions doesn't overwhelm the capability of the network infrastructure. Also, the DBA must proactively plan for growth
#2a	Multi-master environment would allow ICBS databases to replicate DML and DDL amongst the separate ICBS database	The multi-master replication environment is **complex**. Would require increase in resources required both for initial setup but also particularly for ongoing maintenance and administration of the environment Overhead associated with a multi-master replicated environment would be significant particularly given that we will be dealing with 13 participating databases Security: initial setup and ongoing maintenance of multi-master environment would be significant Data consistency: ensuring data consistency across the environment will likely be the greatest source of problems in administering this type of environment Transaction management: the DBA responsible for administering this environment must be careful that the number of transactions doesn't overwhelm the capability of the network infrastructure. Also, the DBA must proactively plan for growth Recovery: in case of failure of one of the database nodes, recovery would not only involve this failed node but also potentially all other participating databases Could potentially require significant investment in network infrastructure in order to implement increased disk mirroring as well as to ensure network redundancy
#2b	Same as #1 Consolidation of separate ICBS databases into single database reduces the complexity of the translation infrastructure Reduction in required administration and maintenance of 13 separate ICBS databases to 1 consolidated ICBS database	Same as #1 Consolidation of the 13 separate ICBS databases into one database would require significant effort, particularly if data structures were not equivalent. Also, impact to other database objects associated with adding new PKs could be significant

Alternative	Advantages	Disadvantages/Risks
	Technically simple solution Databases could be disconnected for extended periods without harm to the read only environment Administration of read only snapshot environment is relatively simple and straightforward	Required modification to current ICBS application could be significant
#3a, #3b	ROSS and ICBS have identical data structures and share data, DML and DDL without the need for translation The design, implementation and maintenance of replicated environments would be simpler Administration and maintenance of the databases themselves would be standardized In the case of #3a, the two databases could be disconnected for extended periods In the case of #3b, data sharing would be real-time	The effort to consolidate and integrate the 13 separate ICBS databases combined with standardizing on a common data structure with ROSS would be significant. This alternative would also require a restructuring of the ROSS data structures. Also, each of these efforts would involve a data migration effort into the new to-be data structures. In addition, in the case of 4b, the data would need to be synchronized i.e. be identical Would likely require a complete rewrite of the ICBS application Would, at the very least, require significant modification of the current ROSS application In the case of #3a, data would not be real-time In the case of #3b, all of the issues outlined in the table for #3a, and #3b regarding the multi-master environment would apply (with the exception that this case would be dealing with only 2 participating databases)
#3c	ROSS and ICBS would be integrated into one single database Administration and maintenance of the environment would potentially be simplified There would be no need to use Oracle replication facilities Data sharing would be real-time	The effort to consolidate and integrate the 13 separate ICBS databases and the ROSS database into a single database would be significant. This effort would require the development of a common data structure as well as a data migration effort into the new to-be data structures. Would require a complete rewrite of the ICBS application Would require significant modification of the current ROSS application

10. Recommendations

10.1 Comparison of Alternatives

As presented in Section 2.1, numerous considerations must be balanced in trying to assess technical alternatives for sharing information between two complex applications. At the heart of such considerations there are three key decision factors: cost, benefits, and risk.

Figure 10-1, below, presents each of the alternatives presented in Section 8, plotted upon a "Cost" vs. "Risk" X/Y axis. "Costs" in this chart are reflective of the implementation cost estimates provided in Appendix C. "Risk" in this chart is reflective of implementation risk due to technical complexity with respect to both application development and establishment of the required COTS technical infrastructure.

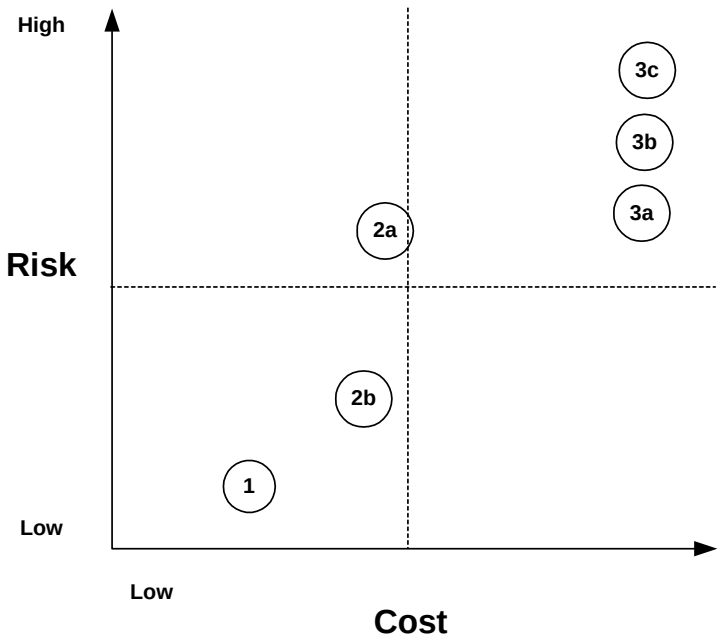


Figure 10-1 ROSS/ICBS Integration Alternatives (Risk versus Cost)

Figure 10-2, below, presents each of the ROSS/ICBS integration alternatives presented in Section 8, plotted upon a "Disruption to User Environment" vs. "Level of ROSS/ICBS Integration" X/Y axis. "Disruption" is based upon the degree of change required by each alternative to the RDBMS infrastructure, data structure, and/or application (ROSS/ICBS) level. "Level of Integration" is based upon the degree to which the respective ROSS and ICBS databases have been combined to achieve real time or near real-time information sharing between ROSS and ICBS communities.

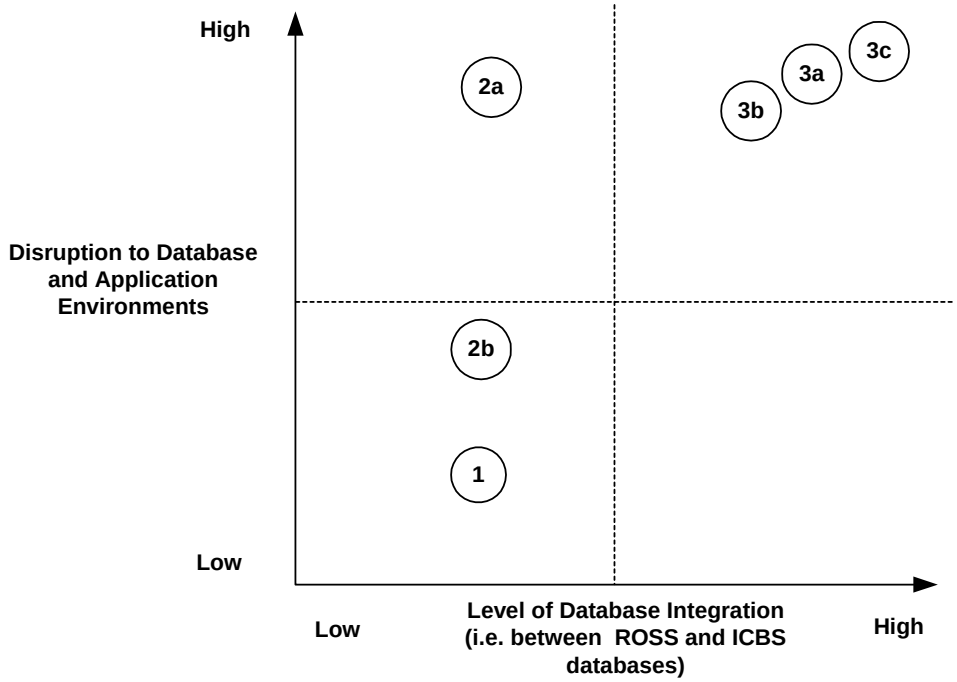


Figure 10-2 ROSS/ICBS Integration Alternatives
(Disruption to Database and Application Environments vs Level of Database Integration)

Intuitively, alternatives presenting high risk and high cost should be avoided, unless such solutions provide benefits that are required by the user community and that cannot be satisfactorily achieved with any other solution.

As clearly demonstrated in Figure 10-1 and Figure 10-2, alternatives #2a and #3a-c, should probably be viewed with disfavor, unless the high cost and high risks associated with such options can be verifiably justified.

The alternatives proposed in this study vary significantly with respect to their impact upon the existing ROSS/ICBS database level and application level infrastructures.

The No Cost alternative, already included in the current ROSS build task, would pose the least disruption to both the ROSS and ICBS environments. However, this alternative offers the least in terms of functionality for it does not provide any meaningful link between the two database environments, and it does not offer any growth path or basis for increased integration between the databases.

Alternatives #1 provides read-write data sharing between ROSS and the ICBS, without significant integration of the existing ICBS cache nodes, nor disruption of the existing ROSS data structures. Depending upon the degree of information sharing required between the two applications, the level of effort expended to develop translation tables and translation logic code can be highly scaled. As such, this alternative presents the lowest risk, with a high degree of modular expandability.

Alternatives #2a would provide real-time data sharing between the individual ICBS databases. However, this alternative requires establishment of a complex multi-master replicated environment with respect to 13 databases. The complexity involved in designing, implementing and maintaining this environment would be substantial. Furthermore, platform and network level infrastructure enhancements to assure high reliability of the participating ICBS database nodes, and to provide sufficient bandwidth to accommodate replication transaction activity, would be significant.

Alternative #2b proposes the integration of the individual ICBS databases into a single database. This alternatives would require substantial modification to the referential integrity of the ICBS data structure (i.e. the use of new primary keys in the affected base tables), and modification of associated table relations, indexes, sequences, and database objects, requiring increased time and effort with respect to database integration.

Alternatives #3a, #3b, and #3c present near-equivalent variations of an integrated ROSS/ICBS "target state." Under Alternatives #3a and #3b, ROSS and ICBS would be redesigned to use identical data structures that would share data via snapshot or multi-master replication. Data sharing would be close to real time, in the case of #3b, or real-time in the case of #3a. Alternative #3c resolves all data replication issues by integrating ROSS and ICBS into a single centralized database with fully integrated data structures.

If selected as initial deployment options, Alternatives #3a, #3b, and #3c, would impose the greatest impact to both the database and application levels of the current ROSS and ICBS environments. However, as discussed in the section that follows, by using a phased approach, the benefits of increased integration can be achieved, over time, in a manner that minimizes impact to operations.

10.2 Recommended Approach

Figure 10-3, below, depicts the recommended phased approach to the integration of ICBS and ROSS. The recommended approach is indicated with solid arrows. "Dashed" arrows are used to show the potential for the use of more complex and more costly technical alternatives, which although not recommended, could be used should operational requirements demand. Design options based on a refined and clearly defined set of requirements will make it possible to control the overall cost and complexity of the effort within each alternative.

10.2.1 Phase I

Alternative #1 - Read Only Snapshots / Translation

Using the "Updateable Snapshots / Translation" as the first development step in the phased approach, it is recommended that the overlapping ROSS/ICBS functions identified in Section 5 be prioritized and that ROSS/ICBS translation capabilities be established for those data structures required to support the highest prioritized capabilities.

10.2.2 Phase II

Alternative #2b - ICBS Consolidation / Updateable Snapshots

The second step in the recommended phased approach path (Figure 10-3) is the consolidation of the remote ICBS caches into a single consolidated ICBS database. If the ROSS and ICBS data structures are to be fully integrated within a single RDBMS, such ICBS consolidation is the next logical step in this progression, allowing data level and application level issues to be identified and resolved with minimal impact to the ICBS operational environment. Translation tables would still be used to transfer information between ROSS and ICBS. However, translation code developed in Phase I would be readily adaptable.

Furthermore, there may be significant life-cycle savings, due to such consolidation, based upon reduced maintenance and support costs. Such savings must be balanced against potential operational concerns, such as the possible operational requirement that the ICBS and ROSS be capable of operating independently in the case of serious loss of wide area network (WAN) connectivity.

Alternate #2a - Multi-Master Updateable Snapshots

If real-time updates (i.e. w/n 1 second or less) are of the utmost importance, transition to a multi-master approach in Phase II could be selected. However, this choice is not favored due the greatly increased complexity and costs with only limited decrease in update time between databases.

10.2.3 Phase III

3a – Identical Schemas / Multi-Master, OR 3b – Identical Schemas / Updateable Snapshots, OR 3c – Integrated Database

The next logical technical step in the progression towards a fully integrated ROSS/ICBS application would be to establish a single, fully integrated ROSS/ICBS database file structure that is commonly and efficiently used by both applications. Such a common data structure could be deployed as a single physical database, or geographically distributed with updates accomplished using Oracle's advanced data replication capabilities. However, such close coupling of the ROSS and ICBS databases should not be executed unless such tight integration is an absolute necessity. Transition to any of the Phase III alternatives will require significant redesign of both the ROSS and ICBS applications.

The impact of full integration can be greatly reduced if the ROSS/ICBS translation tables, developed during Phases I and II, are designed and implemented in accordance with a long-term ROSS/ICBS integrated database design. Over time, each future ROSS and ICBS release could be designed to read/write directly from those segments of the "translation table" which are mature at that time, until operationally appropriate levels of integration have been achieved.

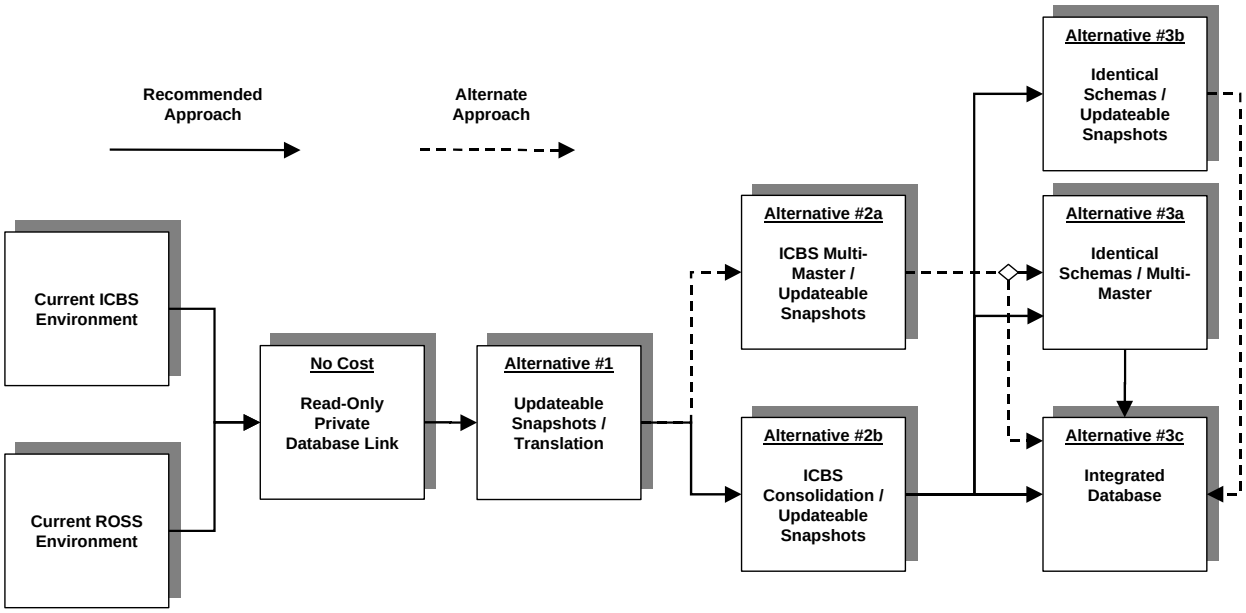


Figure 10-3 Recommended Phased ROSS / ICBS Approach

Appendix A. Work Breakdown Structures - (Removed)

Appendix B. Rough Order of Magnitude (ROM) Cost Estimates - (Removed)

Appendix C. Acronyms

AIX:	IBM proprietary UNIX operating system
VLS:	Versata Logic Server
COTS:	Commercial Off the Shelf
CPU:	Central Processing Unit
DBA:	Database Administrator
DDL:	Data Definition Language
DML:	Data Manipulation Language
DTD:	Data table Definition
GMT:	Greenwich Mean Time
GUI:	Graphical User Interface
ICBS:	Incident Cache Business System
JADE:	Java Development Environment
NFES:	National Fire Equipment System
OLTP:	On-line Transaction Processing
OS:	Operating System
RAID:	Redundant Arrays of Inexpensive Disks
RDBMS:	Relational Database Management System
ROM:	Rough Order of Magnitude
ROSS:	Resource Ordering and Status System
SCSI:	Small Computer Systems Interface
XML:	Extensible Markup Language

Appendix D. Definitions

Authentication Granting access to a system based on supplied credentials.

Backup and Disaster Recovery Plans Documentation describing methodologies which ensure backup of the Oracle database as well as recovery of a database in cases of system failure.

Data Dictionary Listing of all the available tables and columns in a database including table names, column names, column data types, column widths/precision, role of column in referential integrity

Data Integrity **C**haracteristic of data in which the data complies with the relations defined between the tables in the database

Data Quality Characteristic of data in which the actual value or representation of the data element has meaning to the end user and is in a format that complies with the agreed upon format.

Database Link A connection from one database to another that is available to users having proper privileges any time both databases are available.

Data Type Means of characterizing the type of data that a table column can hold.

DDL Data Definition Language Statements are one category of SQL statements. DDL statements define (create) or delete (drop) database objects.

DML Data Manipulation Statements are one category of SQL statements. DML statements, such as select, insert, delete, and update, query and update the actual data in the database.

Export Import Utility An Oracle supplied utility, which provides a method of sharing data among Oracle databases. This utility offers a static means of transferring data.

Extents An allocation of space in the Oracle database which stores information corresponding to a table, index, rollback segment, or temporary object.

Foreign Key A foreign key is a one or more column whose values are based on the primary or candidate key values from another table.

Hot Standby Database One of a number of Oracle high availability solutions in which a secondary database is available should the primary database fail.

Local Wrapper A compiled program, which hides processing, details from the end user.

Multi-Master Replication A table is copied to one or more database, and each database has the ability to insert, update or delete records from it. Modifications are pushed to the other databases at an interval that the DBA sets from each replication group. The highest theoretical frequency is once per second.

National Interagency Incident Cache Business System A system that supports the maintenance of National Interagency depots that house and maintain equipment and supplies for use in local, state and national operations supported by the ROSS system.

Oracle Replication Replication is the process of copying and maintaining database objects, such as tables, in multiple databases that make up a distributed database system. Changes applied at one site are captured and stored locally before being forwarded and applied at each of the remote locations. Oracle replication is a fully integrated feature of the Oracle server; it is not a separate server.

Package A package is a collection of PL/SQL objects that are packaged or grouped together within a special BEGIN-END syntax

Primary Key The primary key is the column(s) used to uniquely identify each row of a table.

Privilege Management The management/administration of the permissions granted to Oracle users allowing the users to execute some action.

Procedural Replication A form of Oracle replication that supports batch processing environments.

Procedure A procedure is a set of instructions (usually combining SQL and PL/SQL commands) saved for calling and repeating execution.

RAID 5 A form of the use of Redundant Arrays of Inexpensive Disks (RAID) in which redundancy is provided by error correction codes (ECCs). Parity information is stored along with the data and is hence stripped across several physical disks.

Read Only Snapshots A master table is copied to one or more databases. Changes in the master table are reflected in the snapshot tables whenever the snapshot refreshes. The snapshot site determines the frequency of the refreshes; data is pulled

Referential Integrity Referential Integrity is the property that guarantees that values from one column depend on values from another column. This property is enforced through integrity constraints.

Schema A schema is a logical grouping of database objects based on the user that owns the objects.

Sequence A sequence is a database object used to generate unique integers for use as primary keys.

SQL*PLUS The Oracle SQL command window.

Synchronous Propagation A means of replicating in which the two-phase commit protocol is utilized to ensure that every transaction is applied at every site.

Transaction Consistency A preservation of the order of transactions when these transactions are applied to sites participating in replication.

Transaction Data Flow The volume of the data transferred as a result of database transactions.

Trigger A database trigger is a stored procedure associated with a table that automatically executes on one or more specified events.

Updateable Snapshots This option is similar to read-only snapshots, except that the snapshot sites are able to modify the data and send their changes back to the master. The snapshot site determines the frequency of the refreshes and the frequency with which updates are sent back to the master.