Siebel Maximum Availability Architecture

With case study on Oracle Private Cloud Appliance and Exadata Database Machine

January 28, 2021, Version 1.0

Public

1. Table of contents

2. Executive Overview

Oracle Maximum Availability Architecture (MAA) is Oracle's best practices blueprint based on proven Oracle high availability technologies and recommendations. The goal of MAA is to achieve the optimal high availability architecture at the lowest cost and complexity.   Papers are published on the Oracle Technology Network (OTN) - http://www.oracle.com/goto/maa.

The Siebel Maximum Availability Architecture is a best practice blueprint for achieving the optimal Siebel high availability deployment using Oracle high availability technologies and recommendations.

In this paper we describe:

- The Siebel MAA architecture along with installation, configuration and operational best practices.
- How Siebel MAA was implemented on Oracle Private Cloud Appliance (PCA) and Exadata.
- Our tests to validate our best practices and measure downtime in various outage scenarios.

When Siebel was configured with our MAA best practices on PCA and Exadata, we demonstrated that there was minimal user impact during typical failure scenarios. In the event of a total site failure the disaster recovery site could be brought online in as little as 10 minutes.

This paper was developed in the Oracle Solutions Center (OSC). The OSC is a centralized global organization with twelve state-of-the-art locations around the world where customers architect, customize and test solutions with Oracle Cloud, Cloud@Customer and On Premises systems in a secure, scalable and interoperable environment for all deployment models. To meet evolving business and technology challenges quickly, OSC provides a wide range of accelerated services that highlight Oracle products and complementary Partner products as needed. Key services include Architecture Review, TCO/ROI Analysis, Proof-of-Concepts, Customized Demonstrations and Workshops to support a dynamic community of VAD/VAR, ISV vendors and System Integrators.

If you are considering High Availability, Disaster Recovery and Datacenter Consolidation using Oracle's Maximum Availability Architecture (MAA), OSC is the place to test, validate and perfect your organization’s Disaster Recovery needs. At the OSC, meet the experts and leverage Oracle's best practices with any combination of technologies hosted by Oracle SMEs and Solution Architects. Contact your local account manager to engage the Oracle Solutions Center and benefit from its range of capabilities and competencies to effortlessly solve your business technology challenges.

3. Introduction

This paper is organized into the following sections:

- A high-level introduction to Oracle Private Cloud Appliance and Oracle Exadata Database Machines
- Siebel Maximum Availability Architecture – a high level description of the architecture and key technology components
- Siebel MAA Case Study on PCA and Exadata – how the MAA architecture was established on our target machines
- Outage Testing and Results – how recovery was performed during various outages
- Summary of Best Practices - a checklist of the best practices recommended by this paper
- Appendix – Case Study State Transitions - detailed steps performed during the case study setup and outage recovery testing

3.1. Introduction to Engineered Systems

Oracle's Engineered Systems combine best-of-breed hardware and software components with game-changing technical innovations. Designed, engineered, and tested to work best together, Oracle's Engineered Systems can power the cloud or streamline data center operations to make traditional deployments even more efficient. The components of Oracle's Engineered Systems are preassembled for targeted functionality and then—as a complete system—optimized for extreme performance. By taking the guesswork out of these highly available, purpose-built solutions, Oracle delivers a solution that is integrated across every layer of the technology stack—a simplicity that translates into less risk and lower costs for your business. Only Oracle can innovate and optimize at every layer of the stack to simplify data center operations, drive down costs, and accelerate business innovation.

4. Oracle Private Cloud Appliance

Oracle Private Cloud Appliance and Oracle Private Cloud at Customer are on-premises cloud native converged infrastructure that allows customers to efficiently consolidate business critical middleware and application workloads. Oracle Private Cloud Appliance is cost effective, easy to manage, and delivers better performance than disparate build-your-
own solutions. Oracle Private Cloud Appliance together with Oracle Exadata provides a powerful, single-vendor, application and database platforms for today's data driven enterprise.

Oracle Private Cloud Appliance runs enterprise workloads alongside cloud-native applications to support a variety of application requirements. Its built-in secure Multi tenancy, zero downtime upgradability, capacity on demand and single pane of glass management make it the ideal infrastructure for rapid deployment of mission critical workloads. Oracle Private Cloud Appliance together with Oracle Cloud Infrastructure provides customers with a complete solution to
securely maintain workloads on both private and public clouds.

4.1. Oracle Exadata Database Machine

Oracle's Exadata Database Machine is Oracle's database platform delivering extreme performance for database applications including Online Transaction Processing, Data Warehousing, Reporting, Batch Processing, or Consolidation of mixed database workloads. Exadata is a pre-configured, pre-tuned, and pre-tested integrated system of servers, networking and storage all optimized around the Oracle database.

5. Siebel Maximum Availability Architecture

Siebel Maximum Availability Architecture (MAA) is a Siebel high availability architecture layered on top of the Oracle Database Maximum Availability Architecture, including a secondary site to provide business continuity in the event of a primary site failure.

In this section we will first present the Oracle Database Maximum Availability Architecture, and then we will describe how to provide high availability for the Siebel application on top of that foundation, resulting in a full Siebel MAA implementation.

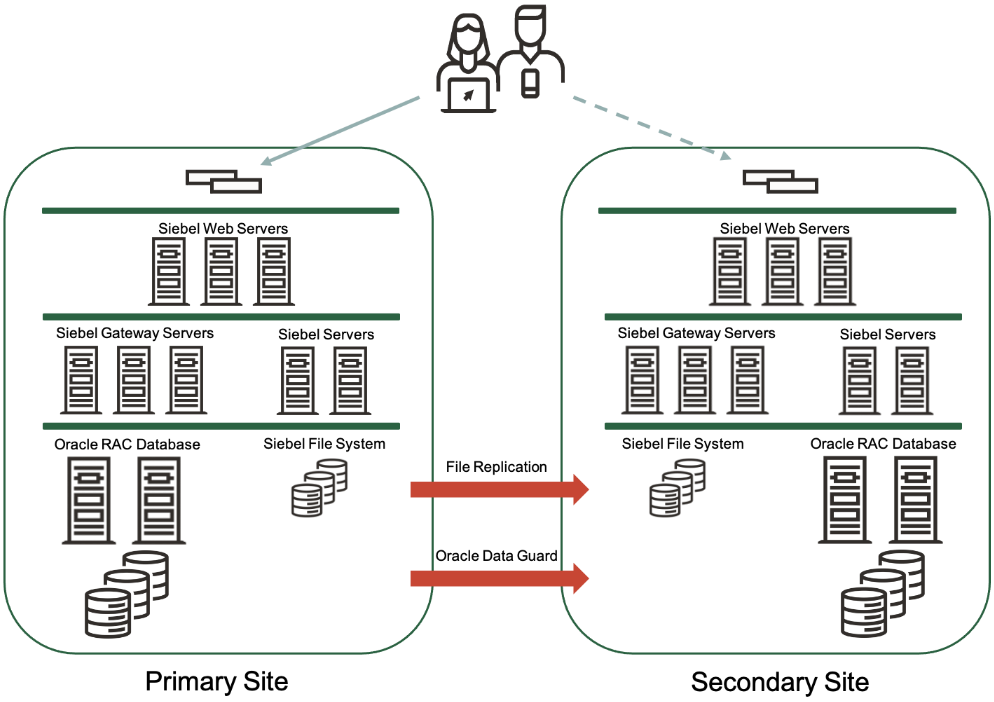

Figure 1 Siebel Maximum Availability Architecture

5.1. Oracle Database Maximum Availability Architecture

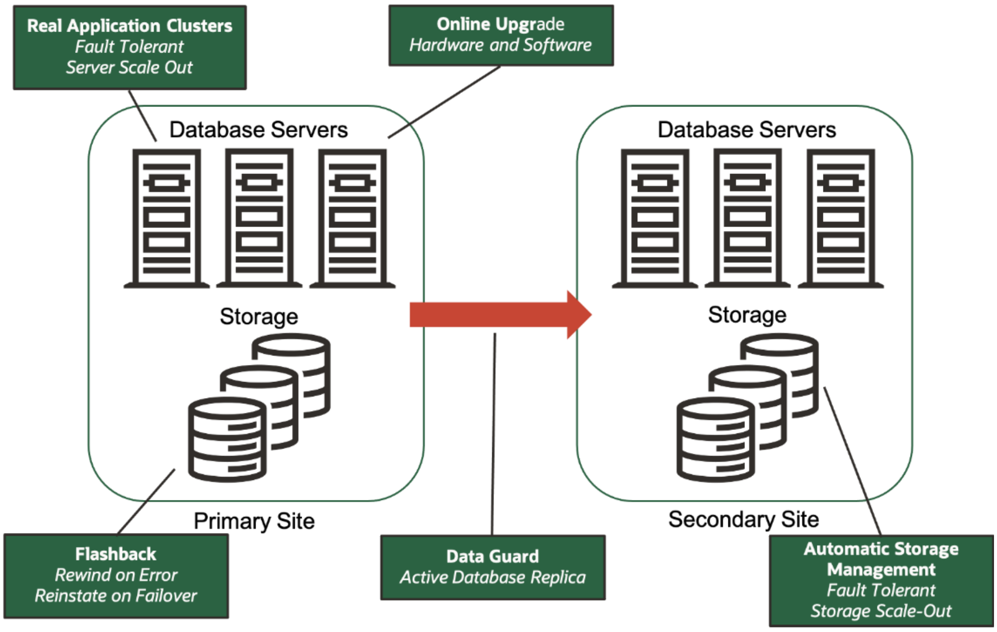

Figure 2 Oracle Database Maximum Availability Architecture

To achieve maximum Siebel application availability, Oracle recommends deploying Siebel on an Oracle Database MAA foundation that includes the following technologies:

- Oracle Real Application Clusters
- Oracle Clusterware
- Oracle Data Guard and Online Upgrade
- Oracle Flashback
- Oracle Automatic Storage Management
- Oracle Recovery Manager and Oracle Secure Backup

We briefly describe each of these technologies in this section. See also: "Oracle Database High Availability Overview" for a thorough introduction to Oracle Database high availability products, features and best practices.

5.1.1. Oracle Real Application Clusters

Oracle Real Application Clusters (RAC) allows the Oracle database to run any packaged or custom application unchanged across a set of clustered nodes. This capability provides the highest levels of availability and the most flexible scalability. If a clustered node fails, the Oracle database will continue running on the surviving nodes. When more processing power is needed, another node can be added without interrupting user access to data. See also: "Oracle Real Application Clusters Administration and Deployment Guide".

5.1.2. Oracle Clusterware

Oracle Clusterware is a general purpose clustering solution originally designed for the Oracle Real Application Clusters active-active multi-instance database and which has been extended to support clustering of all applications. Oracle Clusterware provides traditional HA failover support in addition to online management of protected resources such as online relocation of applications for planned maintenance. Oracle Clusterware is a policy engine providing a rich dependency model for start and stop dependencies, ordered startup and shutdown of applications and defined placement of resources for affinity, dispersion or exclusion. Oracle Clusterware provides a suite of integrated stand-alone or bundled agents for Oracle Application high availability and application resource management.
MAA best practices recommends Oracle Clusterware be used as the clustering solution for the Siebel mid-tier components and the Siebel Bundled Agent for integrated availability and management. See also: "Oracle Clusterware Administration and Deployment Guide".

5.1.3. Oracle Data Guard and Online Upgrade

Oracle Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production Oracle databases to survive failures, disasters, user errors, and data corruption. Data Guard maintains these standby databases as transactionally consistent copies of the production database. If the production database becomes unavailable due to a planned or an unplanned outage, Data Guard can switch any standby database to the production role, thus greatly reducing the application downtime caused by the outage. Data Guard can be used with traditional backup, restore, and clustering solutions to provide a high level of data protection and data availability. See also: "Oracle Data Guard Concepts and Administration".

Siebel supports both physical and logical standby databases. A physical standby database provides a physically identical copy of the primary database, with on disk database structures that are identical to the primary database on a block-for-block basis. A physical standby database is kept synchronized with the primary database through Redo Apply, which recovers the redo data received from the primary database and applies the redo to the physical standby database.

With Active Data Guard, a physical standby database can receive and apply redo while it is open for read-only access and so may be used for other purposes as well as disaster recovery.

A logical standby database contains the same logical information as the production database, although the physical organization and structure of the data can be different. The logical standby database is kept synchronized with the primary database though SQL Apply, which transforms the data in the redo received from the primary database into SQL statements and then executes the SQL statements on the standby database. A logical standby database can be used for disaster recovery and reporting requirements and can also be used to upgrade the database software and apply patch sets while the application is online and with almost no downtime.

With a single command, a physical standby database can be converted into a Snapshot Standby and become an independent database open read-write, ideal for QA and other testing. The Snapshot Standby continues to receive and archive redo data from the primary database while it is open read-write, thus protecting primary data at all times. When testing is complete, a single command will convert the snapshot back into a standby database, and automatically resynchronize it with the primary.

A physical standby database can be used for rolling database upgrades using the SQL Apply (logical standby) process – and return to its function as a physical standby database once the upgrade is complete.

It is possible to deploy a local standby database at the primary site as well as a remote standby at a secondary site. This offers the advantage that a failover to the local standby can be performed while the Siebel Servers continue running - and can be done almost transparently to the end users. It also offers the ability to perform an online database upgrade without the need to switch to another site. We would recommend that both a local and remote standby be deployed for maximum availability.

5.1.4. Oracle Flashback

Oracle Flashback quickly rewinds an Oracle database, table or transaction to a previous time, to correct any problems caused by logical data corruption or user error. It is like a 'rewind button' for your database. Oracle Flashback is also used to quickly return a previously primary database to standby operation after a Data Guard failover, thus eliminating the need to recopy or re-instantiate the entire database from a backup. See MOS ID 565535.1 "Flashback Database Best Practices & Performance", for flashback database best practices.

5.1.5. Oracle Automatic Storage Management

Oracle Automatic Storage Management (ASM) provides a vertically integrated file system and volume manager directly in the Oracle kernel, resulting in:

- Significantly less work to provision database storage
- Higher levels of availability
- Elimination of the expense, installation, and maintenance of specialized storage products
- Unique capabilities for database applications

For optimal performance, ASM spreads files across all available storage. To protect against data loss, ASM extends the concept of SAME (stripe and mirror everything) and adds more flexibility in that it can mirror at the database file level rather than the entire disk level. See also: "Automatic Storage Management".

5.1.6. Oracle Recovery Manager

Recovery Manager (RMAN) is an Oracle database utility that can back up, restore, and recover database files. It is a feature of the Oracle database and does not require separate installation. RMAN integrates with sessions running on an Oracle database to perform a range of backup and recovery activities, including maintaining a repository of historical data about backups. See also: "Oracle Recovery Manager"

5.1.7. Siebel Database Configuration Best Practices

We recommend that Siebel database is configured with the following best practices:

5.1.7.1. Create Role Based Database Services

A database service provides a simple named access point to the database. A service can physically span multiple instances in an Oracle RAC cluster and can be simply moved from one instance to another. By requiring Siebel to connect only through a service we are able to relocate or reconfigure the service without reconfiguring Siebel.

Role-based database services are automatically managed by Oracle Clusterware and are only started when the database is in defined roles.

A database service can be created and configured through Enterprise Manager or using the srvctl command line tool. For example, this is how a service can be created with the srvctl command:

```
srvctl add service -d CDB11 -s SIEB -r "CDB111,CDB112" -j LONG -l "PRIMARY" -y AUTOMATIC
```

PARAMETER	EXAMPLE VALUE	DESCRIPTION
-d	CDB11	The database unique name
-s	SIEB	The database service name
-r	"CDB111,CDB112"	Preferred database instances that will start the service
-l	"PRIMARY"	The database roles in which the service will be automatically started
-j	LONG	Connection load balancing method (Siebel connections normally last a long time)
-l	PRIMARY	The database role under which this service will be started
-y	AUTOMATIC	Indicates that the service should be started automatically

Parameters Used to Add a Role-base Database Service

5.1.7.2. Configure Hugepages (Linux Database Server Only)

Siebel will typically run with many database connections and a large SGA and so configuring hugepages for the Siebel database instances is essential. It is necessary to manually configure sufficient hugepages for the ASM instance and all database instances on each Linux database server node. This will result in more efficient page table memory usage, which is critically important with a large SGA or when there are high numbers of concurrent database connections. Hugepages can only be used for allocating SGA memory space and so do not configure more than is required.

MOS ID 361468.1, “HugePages on Oracle Linux 64-bit” describes how to configure hugepages. Automatic Shared Memory Management (ASMM) can be used with hugepages and so use the SGA_MAX_SIZE parameter to set the SGA size for each instance.

Automatic Memory Management (AMM) cannot be used in conjunction with hugepages and so the MEMORY_TARGET and MEMORY_MAX_TARGET parameters should be unset for each database instance. See MOS ID 749851.1 “HugePages and Oracle Database 11g Automatic Memory Management (AMM) on Linux” for details.

Set the parameter USE_LARGE_PAGES=’only’ for each instance so that the instance will only start if sufficient hugepages are available. See MOS ID 1392497.1 “USE_LARGE_PAGES To Enable HugePages” for details.

It may be necessary to reboot the database server to bring the new hugepages system configuration into effect. Check to make sure that you have sufficient hugepages by starting all the database instances at the same time.

A message is logged to the database alert log when hugepages are being used, for example:

```
****************** Huge Pages Information *****************
Huge Pages memory pool detected (total: 18482 free: 17994)
DFLT Huge Pages allocation successful (allocated: 4609)
***********************************************************
```

In this case, 4609 hugepages were used.

5.1.7.3. Handle Database Password Expiration

By default, Oracle database user passwords will expire after 180 days. Processes should be put in place to refresh passwords regularly or expiration should be extended or disabled. Siebel application availability will be impacted if passwords are allowed to expire. Password expiration for the default user profile can be disabled with the following command:

```
alter profile default limit password_life_time unlimited;
```

5.2. Siebel High Availability Architecture

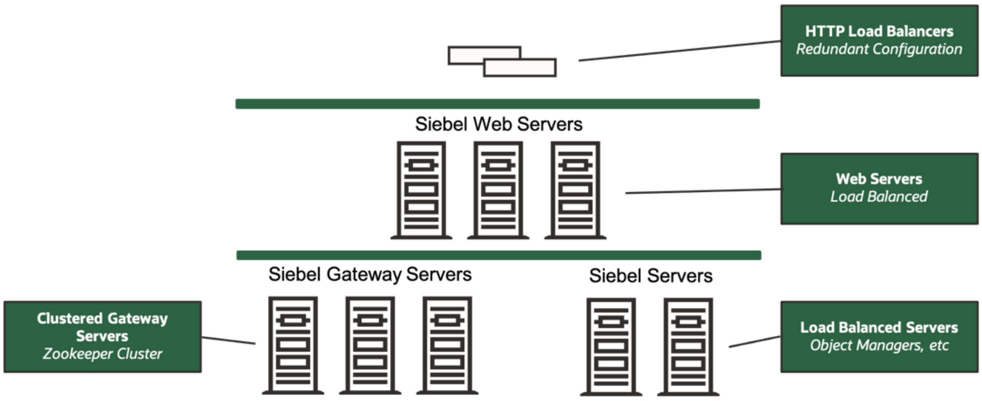

Figure 3 Siebel High Availability Architecture

In this section we discuss the high availability (HA) deployment of the Siebel application software that is layered on top of the Database MAA foundation, as well as the HA requirements for the Siebel file system. See the Siebel Deployment Planning Guide for more details on Siebel HA deployment.

5.2.1. Siebel Application Software HA Deployment Options

Siebel components can each be deployed in a highly available manner, using one of three options depending on the requirements and constraints of the component being deployed – active/active load balanced, distributed services deployed across multiple servers, and active/passive for singletons. In all cases we recommend more than one instance of each component be deployed at each site, on separate physical servers so a server outage does not affect availability. Where more than one instance of a component can be serving users at a time, we recommend the servers have adequate capacity to run peak load even when one server is down.

The four high availability deployment options for Siebel components are:

5.2.1.1. Scalable Services (Load Balancing)

Core Siebel components (e.g., Object Managers, Web Servers) are installed and deployed on multiple servers, and run in an “active/active” configuration for high availability and scalability purposes. Client-initiated workload is distributed across multiple component instances running on multiple servers through load balancing. Web Server load is distributed by an HTTP load balancer. Application Object Manager load is distributed by native Siebel load balancing.

5.2.1.2. Resilient Processing (Distributed Services)

Many Siebel components are implemented as Business Services. Business Services are invoked by other components to complete their business function. In some cases Business Services can be deployed redundantly across multiple Siebel Servers in a configuration known as Distributed Services.

The Siebel Server Request Broker (SRB) balances Service requests across the component instances. In the event that a component instance is lost, the request is re-routed to the surviving instances. An SRB instance will typically be running on all Siebel Servers.

5.2.1.3. Server Clusters

5.2.1.3.1. Cold Failover Server Clusters

Some Siebel services are singletons, meaning only one instance of the service can be running at a time. These are deployed in Siebel Server clusters.

Siebel Server clusters consist of two or more physical servers linked together so that if one server fails, resources such as disks, network addresses, and Siebel components can be switched over to another server. Clustered Siebel components run in an active/passive configuration where a specific Siebel component instance is running on only one physical host at a time. We use Oracle Clusterware (or other 3rd party cluster manager) to monitor and manage the configuration to ensure the components are enabled on only one node of a hardware cluster at a time.

5.2.1.3.2. Native Active-Active Gateway Clusters

Siebel CRM supports an optional native clustering feature for Siebel Gateway to provide high availability benefits to Siebel CRM customers. This feature works at the software level and is the preferred and recommended approach for clustering the Siebel Gateway.

5.2.1.4. Recommended HA Deployment Options

Not all deployment options are supported by all components. The following table gives an example of the supported and preferred options for some of the most commonly deployed components

COMPONENT	CLUSTERING	LOAD BALANCING	DISTRIBUTED SERVICES
Object Manager	Supported (Cold Failover)	Preferred	
Siebel Remote	Preferred		
Workflow Process Manager	Supported		Preferred
Siebel Web Server	Supported	Preferred	
Siebel Gateway Registry	Preferred (Native), Supported (Cold Failover)		
Siebel Gateway Service	Preferred (Native), Supported (Cold Failover)		

Example of HA Deployment Options

5.2.1.5. Deployment on Virtualized Environments

When Siebel components are deployed on virtual machines (VMs) it is important that the physical servers do not become a single point of failure. VM’s of a specific type or component grouping (Web, Object Manager, Gateway, etc) should be arranged across separate physical servers so that a physical server failure will only result in the failure of one instance. This is also known as anti-affinity deployment.

If a physical server fails, it is important that VM’s are restarted on a surviving server or servers as quickly as possible. Some VM managers can be configured to relocate and restart VMs automatically.

5.2.2. Siebel Web Server Load Balancing

It has been found that the optimal web server load balancing is achieved when the following logic is implemented in the HTTP load balancer:

- Load balancing is performed on every request in a round robin fashion with no persistence. This ensures a balanced load across all web servers, smooth recovery when a web server goes down, and almost instant rebalancing when a web server comes up. It is possible to do this with Siebel because session state is not maintained in the Web servers.
- A web server is marked down if attempts to get a static page from the server failed for a period of 16 seconds, checking every 5 seconds. This has been found to be a reliable test of web server availability because it only requires the web server itself to be available to respond to the request and it is unlikely for the web server to be available but not be able to respond within 16 seconds. The load balancer should continue to monitor the downed web server and when it begins to respond it is marked up and requests are once more routed to it.
- If the web server is down then traffic is not routed to that web server, and all existing connections to that server are dropped. If all web servers are marked down then all connections are rejected.
- If, after load balancing, the connection to a web server fails, the other web servers were tried, and only if connections to all web servers fail should an error be returned to the client. A connection failure is most likely due to a web server failure and a precursor to marking the web server down, but this logic prevents many connection errors in the meantime.

5.2.3. Siebel Cluster Deployment

To create a Siebel Cold Failover cluster, you need a cluster manager and a shared Siebel software home.

5.2.3.1. Cluster Manager

- Supports service virtual IP management with failover. The virtual IP address is used as a single network address for the Siebel Server or Gateway Server independent of the physical service location
- Performs service monitoring so it will know when services fail.
- Will restart and relocate Siebel services in the event of failure.

5.2.3.2. Shared Siebel Software Home

- Shared by all cluster nodes for failover.
- Contains Siebel software, remote docking folders, etc.
- Must be deployed in a HA configuration to avoid a single point of failure. Typically, a cluster file system or NFS solution would be used.

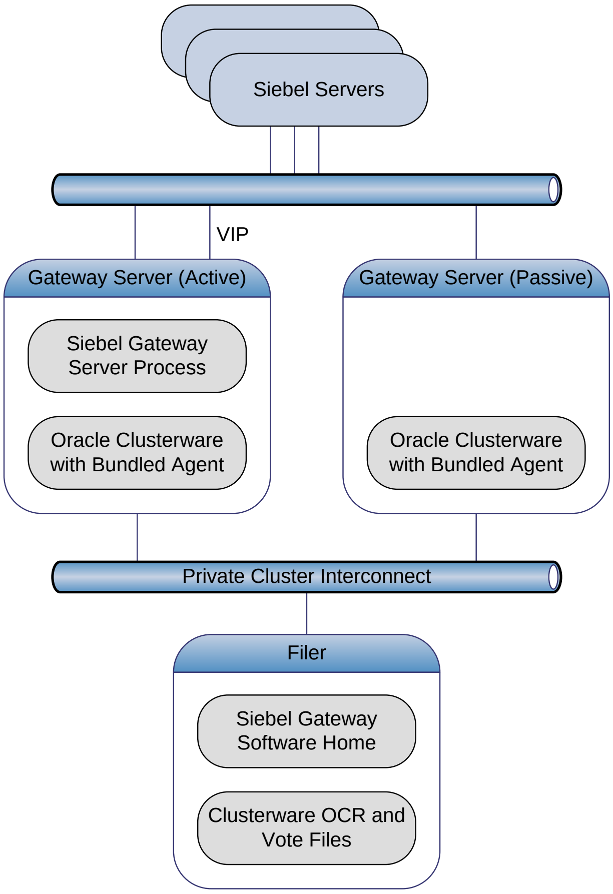
Siebel Servers
VIP
Gateway Server (Active)
Gateway Server (Passive)
Siebel Gateway Server Process
Oracle Clusterware with Bundled Agent
Oracle Clusterware with Bundled Agent
Private Cluster Interconnect
Filer
Siebel Gateway Software Home
Clusterware OCR and Vote Files

Figure 4 Siebel Gateway Server Deployed with Oracle Clusterware and Bundled Agents

Oracle Clusterware may be used as the Cluster Manager for protecting Siebel components and bundled agents are available that are specifically designed to manage and monitor the Siebel Gateway Server and Siebel Servers. For more details see http://www.oracle.com/technetwork/products/clusterware/overview/index.html.

5.2.4. Siebel File System Deployment

The Siebel file system is used to store file attachments and other documents in the Siebel application, and is accessed in parallel by all Siebel Servers. It is a critical part of the Siebel application and so must be deployed in a highly available configuration. Typically, this would be achieved through a cluster file system or network file system (NFS).

The contents of the Siebel File System must be continuously replicated to the secondary site so that the data is available in the event of a primary site failure.

5.2.5. Siebel Tier Configuration Best Practices

We recommend the following when configuring the Siebel application:

5.2.5.1. Database Connection Configuration

When configuring the Siebel application connection to the database it is recommended that the following requirements are met:

- Connection is made through a database service. This is achieved through the SERVICE_NAME parameter in the TNS alias configuration.
- The client must be configured to try all database listeners in the RAC cluster so that a new connection can be established even when nodes are down.
  - If Siebel is configured with an Oracle Database Client at 11g Release 2 or later, and the SCAN feature is configured on the database, then a single SCAN address can be configured, for example: `(ADDRESS=(PROTOCOL=TCP)(HOST=test-scan)(PORT=1521))`

  - Otherwise, each database VIP address should be configured, for example:
    `(ADDRESS=(PROTOCOL=TCP)(HOST=test_vip1)(PORT=1521))`
    `(ADDRESS=(PROTOCOL=TCP)(HOST=test_vip2)(PORT=1521)))`
- The client must be configured to timeout if the connection to a listener is taking too long.
  - If Siebel is configured with an Oracle Database Client at 11g Release 2 or later, the CONNECT_TIMEOUT parameter can be used in the TNS alias configuration, for example:
    `(CONNECT_TIMEOUT=3)`
  - Otherwise, the OUTBOUND_CONNECT_TIMEOUT parameter may be configured in the client side SQLNET.ORA file, for example:
    `SQLNET.OUTBOUND_CONNECT_TIMEOUT=3`
- If Siebel is configured with an Oracle Database Client at 11g Release 2 or later, the RETRY_COUNT parameter may be used to keep retrying the connection, for example:
  `(RETRY_COUNT=3)`

Here is a complete example with an Oracle Database Client 11g Release 2 and SCAN:

```
SIEB = (DESCRIPTION =
  (CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
  (ADDRESS=(PROTOCOL=TCP)(HOST=test-scan)(PORT=1521))
  (CONNECT_DATA= (SERVICE_NAME=SIEB))
)
```

The configuration changes must be the same on all Siebel servers.

5.2.5.2. Reduce TCP Keepalive Timeout

It is possible for some database connections to hang on the Siebel Server if a database node fails, and for TCP connections to hang on the Web Server if the Siebel Server node fails. This is only in the rare case where the node crashes or network fails before the TCP connections can be cleaned up by the operating system. To clean up the "dead" connections it is recommended to reduce the TCP Keepalive Timeout parameters for Siebel Servers, Gateway Servers and Web Servers. Please refer to MOS ID 249213.1 – "Performance problems with Failover when TCP Network goes down (no IP address)" for details on how to configure the TCP Keepalive timeout.

Making these configuration changes may have adverse effect on network utilization and so all changes should be tested and monitored carefully.

5.3. Siebel MAA Site State Model and State Transitions

In the diagram below we picture the states that a deployment goes through as it progresses from the initial single site implementation through the setup, testing and an eventual dual site MAA deployment. The systems will have a specific configuration in each state and there is a set of steps that must be performed to move from one state to the next.

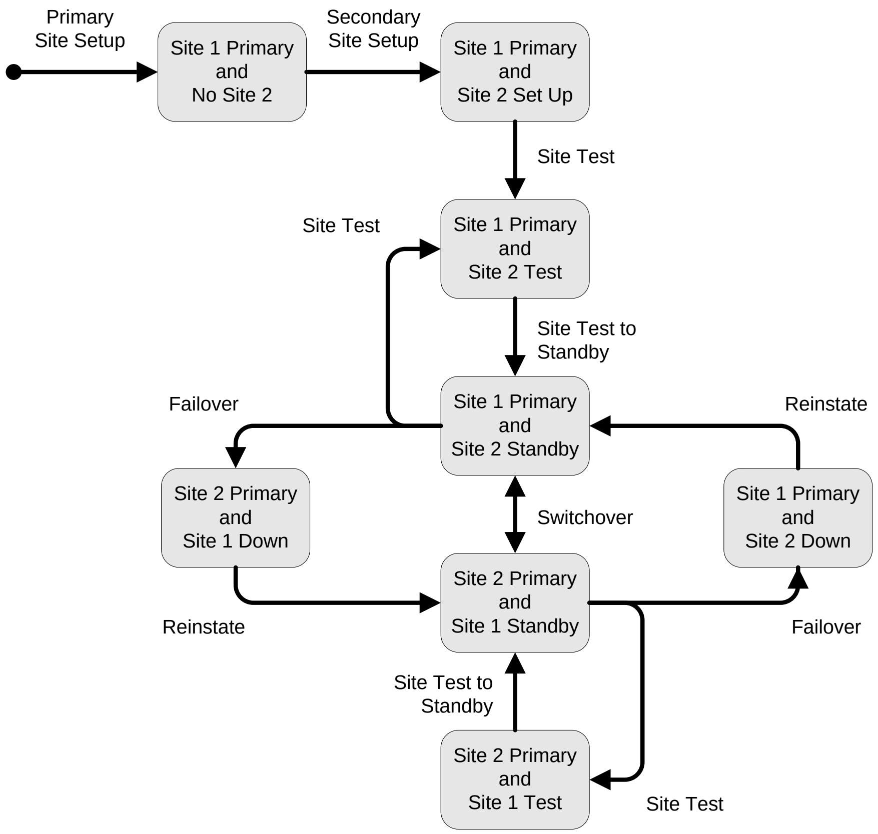
Primary Site Setup
Site 1 Primary and No Site 2
Secondary Site Setup
Site 1 Primary and Site 2 Set Up
Site Test
Site Test
Site 1 Primary and Site 2 Test
Site Test to Standby
Failover
Site 1 Primary and Site 2 Standby
Reinstate
Site 2 Primary and Site 1 Down
Switchover
Site 1 Primary and Site 2 Down
Site 2 Primary and Site 1 Standby
Reinstate
Failover
Site Test to Standby
Site 2 Primary and Site 1 Test
Site Test

Figure 5 Siebel MAA Site State Model and State Transitions

A summary description of the state transitions is provided in the following table:

TRANSITION	DESCRIPTION
Primary Site Setup	Install and configure the primary site.
Secondary Site Setup	Establish the secondary site.
Site Test	Prepare the standby site for a site test.
Site Test to Standby	Convert the site performing a site test back to standby mode.
Switchover	Switch the roles so that the current standby becomes the primary and the current primary becomes the standby.
Failover	Switch the current standby to primary mode. The current primary is assumed to be down or unavailable.
Reinstate Standby	Reinstate the old primary as a standby after failover.

Summary Description of MAA State Transitions

The following table summarizes how the database and Siebel File System are configured in each state:

SITE STATE	SIEBEL DATABASE - DATA GUARD	SIEBEL FILE SYSTEM - REPLICATION
Site 1 Primary and No Site 2	Not configured	Not configured
Site 1 Primary and Site 2 Set Up	Site 1 primary and site 2 physical standby. Snapshot standby during setup.	Site 1 primary with continuous replication to site 2. Site 2 clone during setup.
Site 1 Primary and Site 2 Test	Site 1 primary and site 2 snapshot standby.	Site 1 primary with continuous replication to site 2. Site 2 clone created for test.
Site 1 Primary and Site 2 Standby	Site 1 primary and site 2 physical standby.	Site 1 primary with continuous replication to site 2.
Site 2 Primary and Site 1 Down	Site 2 primary through failover, and site 1 down.	Site 2 primary established from replica, and site 1 down.
Site 2 Primary and Site 1 Standby	Site 2 primary and site 1 physical standby.	Site 2 primary and continuous replication to site 1.
Site 1Primary and Site 2 Down	Site 1 primary through failover and site 2 down.	Site 1 primary established from replica and site 2 down.
Site 2 Primary and Site 1 Test	Site 2 primary and site 1 snapshot standby.	Site 2 primary with continuous replication to site 1. Site 1 clone created for test.

Database and Siebel File System configuration with each state

5.4. Planned and Unplanned Outage Solutions

In the following sections we summarize the outages that may occur in a Siebel environment and the Oracle solution that would be used to minimize application downtime. In all cases, we are focused on Siebel Application downtime as perceived by the end user, not the downtime of the individual component.

5.4.1. Unplanned Outage Solutions

In the following table we describe the unplanned outages that may be caused by system or human failures in a Siebel environment and the technology solutions that would be used to recover and keep downtime to a minimum.

OUTAGE TYPE	ORACLE SOLUTION	BENEFITS	RECOVERY TIME
Siebel Server Node or Component Failure	Load Balancing	Surviving nodes pick up the slack	Affected users reconnect
	Distributed Services	Surviving nodes continue processing	No downtime
	Clustering	Automatic failover to surviving node	Seconds to < 2 minutes
Database Node or Instance Failure	RAC	Automatic recovery of failed nodes and instances. Siebel reconnects automatically.	Users transparently fail over Updates may need to be re-submitted
Site Failure	Data Guard	Fast Start Failover	< 2 minutes [1]
Storage Failure	ASM	Mirroring and automatic rebalance	No downtime
	RMAN with flash recovery area	Fully managed database recovery and disk-based backups	Minutes to hours
	Data Guard	Fast Start Failover	< 2 minutes
Human Error	Oracle Flashback	Database and fine-grained rewind capability	Minutes
	Log Miner	Log analysis	Minutes to hours
Data Corruption	RMAN with flash recovery area	Online block media recovery and managed disk-based backups	Minutes to hours
	Data Guard	Automatic validation of redo blocks before they are applied, fast failover to an uncorrupted standby database	Seconds to 5 minutes

Unplanned Outage Solutions

[1] Site failure will require Siebel Remote re-extraction

5.4.2. Planned Maintenance Solutions

In the following table we summarize the planned maintenance activities that may typically occur in a Siebel environment and the technology solutions that we would recommend to keep downtime to a minimum.

MAINTENANCE ACTIVITY	SOLUTION	SIEBEL OUTAGE
Mid-Tier Operating System or Hardware Upgrade	Siebel Load balancing, distributed services and clustering	No downtime
Siebel Application Patching	Siebel rolling patch application	No downtime
Siebel Application Configuration Change	Siebel Application Restart	Minutes
Siebel Upgrades	Siebel Upgrade and Upgrade Tuner	Hours to days (depending on DB size) [2]
Database Tier Operating System or Hardware Upgrade	Oracle RAC	No downtime
Oracle Database interim patching	Oracle RAC rolling apply	No downtime
Oracle Database online patching	Online Patching	No downtime
Grid Infrastructure upgrade and patches	Rolling apply/upgrade	No downtime
Database storage migration	Oracle ASM	No downtime
Migrating to ASM or migrating a single-instance database to Oracle RAC	Oracle Data Guard	Seconds to minutes
Database patch set and upgrades	Oracle Data Guard logical standby	Seconds to minutes

Planned Maintenance Solutions

[2] In reality there are a number of ways to mitigate the impact of extended upgrade downtime, for example, by providing a read-only replica. Oracle Consulting Services have significant experience in this area and can help to plan and execute the upgrade.

6. Siebel MAA Case Study on Oracle PCA and Exadata

In this section we describe how the Siebel Maximum Availability Architecture described in the earlier chapters was deployed on a system consisting of PCA and Exadata machines. Oracle Site Guard was also installed and configured to manage the site transitions. A high-level view of the configured system is pictured below:

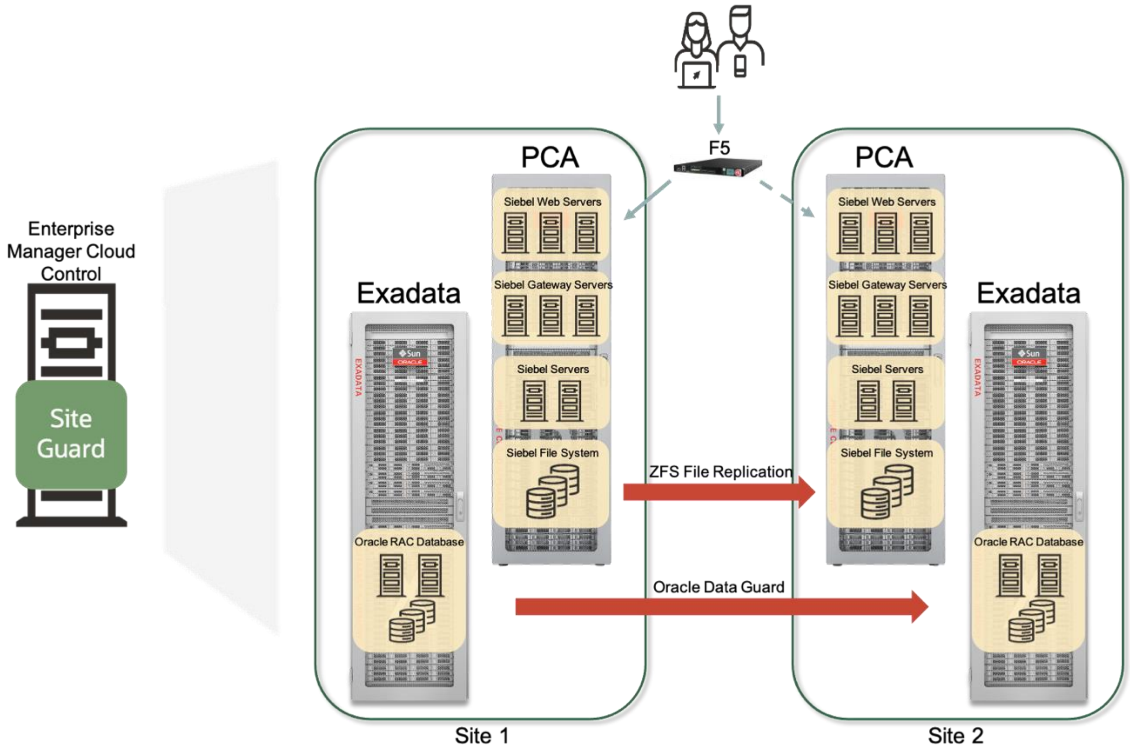

Figure 6 Case Study System Configuration

6.1. Systems

The following systems were used for the case study:

SYSTEM	SITE 1	SITE 2
F5	F5 BIG-IP 4200	
PCA	X8-2	X5
Exadata	X7-2 Quarter Rack	X7-2 Quarter Rack

6.2. Software

The following software was used for the case study:

SOFTWARE	VERSION
PCA	2.4.3
Siebel	20.10
Exadata Database	19.8
Exadata Grid Infrastructure	19.8

Database Client	12.2
Exadata	20.1.1
F5 BIG-IP Local Traffic Manager	15.1.0 Build 0.0.31

6.3. State Transitions

The detailed steps that were followed to setup and test the system are documented in Appendix – Case Study State Transitions:

- Primary Site Setup
- Secondary Site Setup
- Site Test
- Site Test to Standby
- Switchover
- Failover
- Reinstate Standby

The transitions were executed manually and in addition, Oracle Site Guard was installed on configured to perform the Switchover and Failover transitions. Please see the Technical Brief entitled “Application-Level Disaster Recovery using Site Guard” for details on how to configure Site Guard and the Siebel transitions.

Scripts were created to start up and shutdown the Siebel application both for manual and Site Guard automated transitions. For resilience, the scripts were implemented on each Siebel Gateway Server.

SCRIPT NAME	IMPLEMENTED ON	PURPOSE
/home/oracle/prim_up	Site 1 Siebel Gateway Servers	Site 1 Siebel Start Up
/home/oracle/prim_down	Site 1 Siebel Gateway Servers	Site 1 Siebel Shutdown
/home/oracle/sec_up	Site 2 Siebel Gateway Servers	Site 2 Siebel Start Up
/home/oracle/sec_down	Site 2 Siebel Gateway Servers	Site 2 Siebel Shutdown

6.4. Oracle Private Cloud Appliance

The Siebel application servers were all deployed on the PCA.

6.4.1. Shared File Systems on ZFS Filers

The Siebel File System was hosted on the highly available ZFS Storage Appliances located in the PCA machines. The following should be considered when creating shared file systems, replicas, and clones:

6.4.1.1. Considerations When Creating Shared File Systems

- Sufficient quota should be allocated to support the projected size of the file system.
- The file system should be created in a separate project to provide independent replication to the secondary site.
- Network. Read/write and root access must be granted to the Siebel Servers.

- The Siebel owning user and group (oracle 54321 and oinstall 54321 in our case study) must be given root directory access.

6.4.1.2. Considerations when Creating a Remote Replica

- Project Level replication is recommended. It is a requirement for Switchover and Failover transitions performed by Oracle Site Guard.
- Continuous replication is recommended for the Siebel File System.

6.4.1.3. Considerations When Creating a Clone of a File System Replica

- Use the "Clone most recently received project snapshot" icon (labeled with the + sign)
- Make sure that the cloned file system has a distinct project name, name and mount point (override) so that it will not be confused with the production file system.

6.4.2. Application Deployment

The following application components were deployed for the case study:

SYSTEM	SITE 1 VM COUNT	SITE 2 VM COUNT
Siebel Web Server	2	2
Siebel Server	2	2
Siebel Gateway Server	3	3

The deployment of Siebel components in a Cold-Failover Cluster was out of scope for this case study. A case study on how this can be done for an earlier version of Siebel is available in this paper - Siebel MAA with Case Study on Exalogic and Exadata.

6.5. Exadata

The Siebel database was hosted on Exadata on site 1 and site 2. The Exadata machines had a standard configuration.

6.5.1. Database Services

The SIEB database service was created on the primary and standby to provide access to the database when in primary mode.

6.5.2. Exadata Exachk Best Practices

Exachk was run after the database was created to validate the configuration.

6.6. F5

F5 Networks BIG-IP Local Traffic Manager (LTM) application delivery controllers were used for Siebel Web Server load balancing on the primary and secondary sites. The BIG-IP LTM provides application health monitoring, TCP connection management, load balancing, and high availability to the Siebel Web tier.

The base configuration of the F5 BIG-IP was done in the accordance with the existing "F5 Siebel 8.0 Deployment Guide".

7. Site Outage Testing and Results

All servers on the primary site were stopped abruptly and site failover was performed by Site Guard. All application users failed on site outage and the site failover procedure was followed to restore service on the standby site. The failover procedure was performed by Site Guard and the timing for each step are shown in the following table:

OPERATION	TYPICAL ELAPSED TIME
Failover Storage	1m 6s
Mount Siebel File System	18s
Database Failover	3m 0s
Start Siebel Startup (Post-Scripts)	40s
Wait for first Siebel Login	3m 4s
TOTAL	8m 8s

Unplanned Site Outage Timings

8. Summary of Best Practices

Here is a summary of the best practices that have been presented in this paper providing a checklist for a Siebel MAA implementation.

8.1. Best Practices Siebel Database High Availability

Here are the Siebel database best practices that should be applied to the primary and secondary site to achieve highest availability:

- Deploy Siebel on an Oracle RAC database for the highest availability and scalability.
- Use Automatic Storage Management to simplify the provisioning and management of database storage.
- Enable Oracle Flashback Database to provide the ability to "rewind" the database in the event of user errors.
- Use Oracle Recovery Manager to regularly backup the Siebel database.
- Always use Hugepages for Siebel databases on Linux. Monitor memory usage and adjust the workload and parameters accordingly.
- Revalidate the configuration regularly and especially after changes are made. Exachk can be used to assist in the validation process when deployed on Exadata.

8.2. Best Practices for Siebel Application High Availability

Here are the Siebel application best practices that should be applied to the primary and secondary site to achieve highest availability:

- Deploy multiple Siebel servers and deployed all critical Siebel components in a load balanced, distributed service, or clustered configuration, so that work can continue in the event of a Siebel Server node failure.
- Deploy multiple web servers so that work can continue normally if there is a web server outage.
- Deploy a load balancer in a redundant configuration and load balance web server load using our recommended logic.
- Deploy the Siebel File System on a fault tolerant filer.
- Take regular backups of the Siebel Servers, Web Servers, Gateway Servers, and Siebel File System.
- Take regular backups of the Siebel Gateway Server backing file using the Siebel server manager.
- Connect to the database through the role-based services by connecting through all the possible database listeners and with connection timeouts and retries.
- Reduce TCP Keepalive Timeout on Siebel Web Servers, Siebel Servers, and Siebel Gateway Servers.

8.3. Best Practices for Disaster Readiness and Recovery

Here are the best practices for deploying a secondary site and recovery procedures in readiness for a site outage:

- Deploy a second geographically separated site that can run the Siebel workload in the event the primary site is down.
- Use Data Guard to replicate all database changes to a standby database located on the secondary site.
- Take advantage of Active Database Guard to offload read-only queries to the standby database.
- Enable Oracle Flashback Database so the old primary database can be quickly reinstated as a standby in the event of site failover.
- Continuously replicate the Siebel File System to the secondary site with minimal lag. Develop procedures for how to reverse the direction of replication in the event of failover or switchover, and procedures to clone the replica for site testing.

- Export the Siebel File System primary, standby replica, and clones, with different names to avoid mounting the incorrect one.
- Develop and document operational procedures in line with the Siebel MAA state model and state transitions.
- Use Data Guard Broker to simplify Data Guard administration.
- Use Oracle Site Guard to automate site switchovers and failovers.
- Use the snapshot standby to provide an updatable replica of the primary database for temporary site testing.

9. Appendix – Case Study State Transitions

In this appendix we document the detail the procedures that were followed during the case study setup and outage testing. They implement the state transitions as described in the section - Siebel MAA Site State Model and State Transitions.

9.1. Primary Site Setup

9.1.1. Siebel Database Setup

The standard Exadata configuration was deployed on the primary site.

9.1.1.1. Configure Hugepages

Hugepages were configured. It is critically important that hugepages are configured when running Siebel databases on Linux database platforms.

9.1.1.2. Siebel Database Creation

The Siebel database was copied over from another test system using RMAN duplicate functionality. The database was a pluggable database (PDB1) within a container database (CDB11).

9.1.1.3. OS Environment

The oracle user OS environment was configured on each database server, for example:

```
cat >siebel.env <<'EOF'
export ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_1
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_SID=CDB111
EOF

cat >grid.env <<'EOF'
export ORACLE_HOME=/u01/app/19.0.0.0/grid
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_SID=+ASM1
EOF
```

9.1.1.4. Database Registration

The database was registered in Oracle Clusterware as follows:

```
srvctl add database -db CDB11 -oraclehome /u01/app/oracle/product/19.0.0.0/dbhome_1
srvctl add instance -db CDB11 -i CDB111 -n exa14db01
srvctl add instance -db CDB11 -i CDB112 -n exa14db02
srvctl modify database -db CDB11 -a "DATAC1,RECOC1"
```

9.1.1.5. Database Service Creation

```
srvctl add service -d CDB11 -s SIEB -r "CDB111,CDB112" -j LONG -l "PRIMARY" -y AUTOMATIC -pdb
PDB1
```

9.1.1.6. Run Exachk

Exachk was run after the database was created to validate the configuration.

9.1.2. Shared File System Creation

File systems were created in the PCA ZFS Storage Appliance for site 1 as follows:

PURPOSE	SHARE TYPE	EXPORTED AS
Siebel File System	Local	/export/siebelfs

Please see the section Considerations When Creating Shared File Systems for the additional considerations when creating the file system.

9.1.3. Siebel Virtual Machine Creation

VMs were created on the PCA with Oracle Linux 7, an appropriate size, High Availability enabled, and assigned to an Anti-Affinity Group. The following table summarizes the settings:

PURPOSE	COUNT	CPU	RAM	STORAGE	ANTI-AFFINITY GROUP
Siebel Gateway	3	16	32	500GB	SiebGtwy
Siebel Web	2	16	32	500GB	SiebWeb
Siebel Server	2	16	32	500GB	SiebSrvr

When creating the VMs it was important to configure interfaces to allow access to the database servers, ZFS storage, clients and administrators. Note, the Siebel Web Server VMs do not require access to the storage network.

The VMs that were created were as follows:

PURPOSE	NUMBER OF VMS
Siebel Gateway Servers	3
Siebel Servers	2
Siebel Web Servers	2

After creation, each VM was configured as follows:

- Hostname (/etc/sysconfig/network)
- DNS (/etc/resolv.conf)
- NTP was configured to synchronize the system clock
- TCP Keepalive Parameters were configured as follows on each vServer:

```
net.ipv4.tcp_keepalive_time = 60
net.ipv4.tcp_keepalive_probes = 6
net.ipv4.tcp_keepalive_intvl = 10
```

- Software packages installed:

```
yum -y install libgpg-error-devel
yum -y install compat-libstdc++-33
yum -y install libXau.i686
yum -y install libXdmcp.i686
yum -y install libX11.i686
yum -y install libXext.i686
yum -y install libXtst.i686
yum -y install libXi.i686
yum -y install libXp.i686
yum -y install libXpm.i686
yum -y install glibc
yum -y install glibc-devel.i686
yum -y install libaio.i686
yum -y install libstdc++.i686
yum -y install libaio-devel.i686
yum install libstdc++44-devel.i686
yum -y install libstdc++
yum -y install libstdc++-devel.i686
yum -y install ksh
```

- Oracle Instant Client packages installed:

```
oracle-instantclient12.2-basic-12.2.0.1.0-1.i386.rpm
oracle-instantclient12.2-sqlplus-12.2.0.1.0-1.i386.rpm
```

- Oracle User Created and Configured:

```
groupadd -g 54321 oinstall
useradd -u 54321 -d /home/oracle -g oinstall -m oracle
echo "<password>" | passwd oracle –stdin

cat - >/home/oracle/.bashrc <<!
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
       . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
export ORACLE_HOME=/usr/lib/oracle/12.2/client
export LD_LIBRARY_PATH=\$ORACLE_HOME/lib
export PATH=$PATH:\$ORACLE_HOME/bin
export RESOLV_MULTI=off
!

chmod 644 /home/oracle/.bashrc
chown oracle:oinstall /home/oracle/.bashrc
```

- Instant Client Configured:

```
export CLIENT_HOME=/usr/lib/oracle/12.2/client
cd $CLIENT_HOME /lib
ln -s libclntsh.so.12.1 libclntsh.so
mkdir -p $CLIENT_HOME/network/admin
cat - >$CLIENT_HOME/network/admin/tnsnames.ora <<!
SIEB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan1)(PORT = 1521))
    (CONNECT_TIMEOUT=5)
    (TRANSPORT_CONNECT_TIMEOUT=3)
    (RETRY_COUNT=3)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SIEB)
    )
  )
!
chown oracle:oinstall $CLIENT_HOME/network/admin/tnsnames.ora
```

9.1.4. File System Folders Created

Software folders were created as follows for site 1:

PURPOSE	FOLDER NAME	LOCATION
Oracle Inventory	/u01/app/oraInventory	All Servers
Siebel File System	/siebelfs	All Siebel Servers
Siebel Software Home	/u01/app/oracle/product/siebelhome/ses	All Servers

All the above folders were owned by the user oracle and group oinstall.

9.1.5. Shared File Systems Mounted on Siebel Servers

Shared file systems were mounted as follows:

PURPOSE	MOUNTED ON	EXPORTED AS (MOUNTED ON)	MOUNT OPTIONS
Siebel File System	All Siebel Servers	/export/siebelfs (/siebelfs)	_netdev,hard,intr,noatime,rsize=32768,wsize=32768

9.1.6. Siebel Installation and Configuration

Siebel 17.0 was installed and Siebel update 20.10 was applied on each Siebel VM as per the Siebel documentation. The three gateway VMs were used to create a Siebel Gateway Cluster.

9.1.7. Web Server Load Balancing Configuration

The web server load balancing was configured in the F5 load balancer. The configuration was performed in accordance with the second chapter of the F5 "F5 Siebel Deployment Guide".

- HTTP health monitor:

PROPERTY	VALUE

Interval	5
Timeout	16
Send String	GET /index.html \r\n

- Pool:

PROPERTY	VALUE
Load Balancing Method	Round Robin
Action on Service Down	Reject

- Virtual Server:

PROPERTY	VALUE
Default Persistence Profile	None

9.2. Secondary Site Setup

In this case study the secondary site was located on PCA and Exadata. The following steps were performed to create the secondary site:

9.2.1. Establish Standby Database

9.2.1.1. Database Server Setup

The standard Exadata configuration was deployed on the secondary site with disk groups +DATAC2 and +RECOC2.

It was decided to use the following naming for the database and instances on the secondary site:

SITE	DB_UNIQUE_NAME	DB_NAME	INSTANCES
1 (primary)	CDB11	CDB11	CDB111 on exa14db01, and CDB112 on exa14db02
2 (secondary)	S2CDB11	CDB11	CDB111 on exa14db03, and CDB112 on exa14db04

9.2.1.2. Oracle Home Installation

A database software home (Oracle home) was installed on the secondary site database server nodes with the following attributes:

- Created under the same username oracle and with the same folder name as the primary for simplicity.
- Same version and patch level as the primary.

The oracle OS environment was configured on each database server, for example:

```
cat >siebel.env <<'EOF'
export ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_1
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_SID=CDB111
EOF

cat >grid.env <<'EOF'
export ORACLE_HOME=/u01/app/19.0.0.0/grid
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_SID=+ASM1
EOF
```

9.2.1.3. ASM Folder Creation

The database folders were created in the +DATAC2 disk group, as user grid on scam08db03:

```
. grid.env
asmcmd <<EOF
mkdir +DATAC2/S2CDB11
mkdir +DATAC2/S2CDB11/DATAFILE
mkdir +DATAC2/S2CDB11/ONLINELOG
mkdir +DATAC2/S2CDB11/CONTROLFILE
mkdir +DATAC2/S2CDB11/PASSWORD
mkdir +DATAC2/S2CDB11/PARAMETERFILE
mkdir +DATAC2/S2CDB11/TEMPFILE
EOF
```

9.2.1.4. Init.ora File Creation

Init.ora files were created on all the site 2 nodes, for example as oracle on exa14db03:

```
. siebel.env
cd $ORACLE_HOME/dbs
cat >initCDB111.ora <<'EOF'
spfile='+DATAC2/S2CDB11/PARAMETERFILE/spfile.ora'
EOF
```

And, as oracle on exa14db04:

```
. siebel.env
cd $ORACLE_HOME/dbs
cat >initCDB112.ora <<'EOF'
spfile='+DATAC2/S2CDB11/PARAMETERFILE/spfile.ora'
EOF
```

9.2.1.5. Initial Database Parameters

The database parameters were configured on the standby with the same considerations as on the primary. To simplify the configuration process, the parameters were extracted from the primary, copied to the secondary, edited, and then a spfile was created on the secondary site. As oracle on exa14db01:

```
. siebel.env
cd $ORACLE_HOME/dbs
sqlplus / as sysdba <<EOF
create pfile='standby_params' from spfile;
EOF
scp standby_params oracle@exa14db03:$ORACLE_HOME/dbs/
```

The Data Guard parameters necessary for establishing a standby database were considered separately and later on in the configuration process. The following parameters were found to need different settings on the standby:

```
CDB111.cluster_interconnects='192.168.61.5:192.168.61.6'
CDB112.cluster_interconnects='192.168.61.7:192.168.61.8'
*.control_files='+DATAC2/S2CDB11/CONTROLFILE/current.646.1052310485'
*.db_create_file_dest='+DATAC2'
*.db_create_online_log_dest_1='+DATAC2'
*.db_recovery_file_dest='+RECOC2'
```

As oracle on exa14db03 edit the parameter file:

```
. siebel.env
vi $ORACLE_HOME/dbs/standby_params
```

Then create the spfile:

```
sqlplus / as sysdba <<EOF
create spfile='+DATAC2/S2CDB11/PARAMETERFILE/spfile.ora' from pfile='$ORACLE_HOME/dbs
/standby_params';
EOF
```

9.2.1.6. Database Networking Configuration

The following aliases were added to the tnsnames.ora in each Oracle home on site 1 and site 2 to allow Data Guard service and broker connectivity between sites:

```
dg_cdb11 =
  (DESCRIPTION =
    (SDU=65536) (RECV_BUF_SIZE=134217728)
    (SEND_BUF_SIZE=134217728)
    (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = cdb11)
      (UR=A)
    )
  )

dg_s2cdb11 =
  (DESCRIPTION =
    (SDU=65536) (RECV_BUF_SIZE=134217728)
    (SEND_BUF_SIZE=134217728)
    (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = s2cdb11)
      (UR=A)
    )
  )
```

9.2.1.7. Enable Archive Log Mode, Flashback Database and Force Logging on Primary

Archive log mode, flashback database and force logging were enabled on the primary, as user oracle on exa14db01:

```
. siebel.env
srvctl stop database -db cdb11 -stopoption IMMEDIATE

sqlplus / as sysdba <<EOF
startup mount
alter database archivelog;
alter database flashback on;
alter database open;
alter database force logging;
EOF

srvctl start database -db cdb11
```

9.2.1.8. Establish Password Files

A password file was created and copied to all other nodes on the primary and secondary sites.

As oracle on exa14db01:

```
. siebel.env
orapwd file='+DATAC1/CDB11/PASSWORD/orapw' password='<password>' dbuniquename='CDB11'
force=true

. grid.env
asmcmd pwget --dbuniquename CDB11

. grid.env
asmcmd pwcopy +DATAC1/CDB11/PASSWORD/orapw /home/oracle/orapw

. siebel.env
scp /home/oracle/orapw oracle@exa14db03:orapw
```

As oracle on exa14db03:

```
. grid.env
asmcmd pwcopy --dbuniquename S2CDB11 -f /home/oracle/orapw +DATAC2/S2CDB11/PASSWORD/orapw

. grid.env
asmcmd pwget --dbuniquename S2CDB11
```

9.2.1.9. Create Standby Redo Logs on Primary

The following was executed as user oracle on exa14db01 to determine the number and size of redo logs:

```
. siebel.env
sqlplus / as sysdba <<'EOF'
select thread#, count(*) log_count, max(bytes) log_size from v$log group by thread#;
EOF
```

The output was:

```
   THREAD#  LOG_COUNT   LOG_SIZE
---------- ---------- ----------
         1          4 4294967296
         2          4 4294967296
```

Standby redo logs were created of the same size for each log file in each thread:

```
sqlplus / as sysdba <<'EOF'
alter database add standby logfile thread 1 size 4294967296;
alter database add standby logfile thread 1 size 4294967296;
alter database add standby logfile thread 1 size 4294967296;
alter database add standby logfile thread 1 size 4294967296;

alter database add standby logfile thread 2 size 4294967296;
alter database add standby logfile thread 2 size 4294967296;
alter database add standby logfile thread 2 size 4294967296;
alter database add standby logfile thread 2 size 4294967296;
EOF
```

9.2.1.10. Configure Data Guard Parameters on Primary

The following was executed as user oracle on exa14db01 (the instance was assumed to be running):

```
sqlplus / as sysdba <<'EOF'
alter system set dg_broker_start=false scope=both sid='*';
alter system set log_archive_config='dg_config=(cdb11,s2cdb11)' scope=both sid='*';
alter system set log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST' scope=both sid='*';
alter system set standby_file_management='AUTO' scope=both sid='*';
alter system set DG_BROKER_CONFIG_FILE1='+DATAC1/cdb11/dg1.dat' scope=both sid='*';
alter system set DG_BROKER_CONFIG_FILE2='+RECOC1/cdb11/dg2.dat' scope=both sid='*';
EOF
```

9.2.1.11. Configure Data Guard Parameters on Secondary

The following was executed as user oracle_siebel on exa14db03:

```
. siebel.env
sqlplus / as sysdba <<'!'
startup nomount force;
alter system set dg_broker_start=false scope=both sid='*';
alter system set db_unique_name='s2cdb11' scope=spfile sid='*';
alter system set log_archive_config='dg_config=(cdb11,s2cdb11)' scope=both sid='*';
alter system set log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST' scope=both sid='*';
alter system set standby_file_management='AUTO' scope=both sid='*';
alter system set DG_BROKER_CONFIG_FILE1='+DATAC2/s2cdb11/dg1.dat' scope=both sid='*';
alter system set DG_BROKER_CONFIG_FILE2='+RECOC2/s2cdb11/dg2.dat' scope=both sid='*';
shutdown immediate
!
```

9.2.1.12. Instantiate the Standby Database

There are a number of different ways to instantiate the standby database from the primary. In this case restore from service was used. As user oracle on exa14db03:

```
. siebel.env
rman target / <<!
startup nomount force;
restore standby controlfile from service dg_cdb11;
alter database mount;
CONFIGURE DEVICE TYPE DISK PARALLELISM 4;
restore database from service dg_cdb11 section size 64G;
shutdown immediate
!
```

9.2.1.13. Clear the Logs

```
. siebel.env
sqlplus / as sysdba <<'!'
set pagesize 0 feedback off linesize 120 trimspool on
spool /tmp/clearlogs.sql
select distinct 'alter database clear logfile group '||group#||';' from v$logfile;
spool off
@/tmp/clearlogs.sql
select member from v$logfile;
!
```

9.2.1.14. Register the Database in Oracle Clusterware

```
. siebel.env
srvctl add database -d s2cdb11 -n cdb11 -o $ORACLE_HOME
srvctl add instance -d s2cdb11 -i "CDB111" -n exa14db03
srvctl add instance -d s2cdb11 -i "CDB112" -n exa14db04
srvctl modify database -d s2cdb11 -a "+DATAC2,+RECOC2"
```

9.2.1.15. Create Database Services

As user oracle on exa14db03:

```
srvctl add service -d s2cdb11 -s SIEB -r "CDB111,CDB112" -j LONG -l "PRIMARY " -y AUTOMATIC -
pdb PDB1
```

9.2.1.16. Mount Remaining Standby Instances

As user oracle on exa14db03:

```
srvctl start database -db s2cdb11 -startoption MOUNT
```

9.2.1.17. Configure and Start Data Guard Broker

On Primary and Standby as user oracle:

```
sqlplus / as sysdba <<EOF
alter system set dg_broker_start=true scope=both sid='*';
EOF
```

On the primary, as user oracle:

```
dgmgrl sys/'<password>'@dg_cdb11
create configuration config as primary database is cdb11 connect identifier is dg_cdb11;
add database s2cdb11 as connect identifier is dg_s2cdb11;
enable configuration;
```

9.2.1.18. Validate Standby Operation

As user oracle on any database server:

```
. siebel.env
dgmgrl sys/'<password>'@dg_s2cdb11 <<EOF
show configuration
show database cdb11
show database s2cdb11
EOF
```

9.2.1.19. Enable Flashback Database on Standby

As user oracle on exa14db03:
. siebel.env
srvctl stop database -db s2cdb11
srvctl start database -db s2cdb11 -startoption MOUNT
sqlplus / as sysdba <<EOF

9.2.1.20. Customize the Broker Parameters

As user oracle on any database server:

```
dgmgrl sys/'<password>'@dg_cdb11 <<EOF
edit database s2cdb11 set property LogArchiveMaxProcesses=5;
edit database cdb11 set property LogArchiveMaxProcesses=5;
edit database s2cdb11 set property StandbyFileManagement=auto;
edit database cdb11 set property StandbyFileManagement=auto
EOF
```

9.2.1.21. Start the Database in Snapshot Standby Mode

The standby database was started in snapshot standby mode to complete and test the middle tier configuration. As user oracle on any database server:

```
dgmgrl sys/'<password>'@dg_s2cdb11 <<EOF
convert database s2cdb11 to snapshot standby
EOF
```

The standby was periodically converted back to physical standby mode to apply any accumulated redo. As user oracle_siebel on any database server:

```
dgmgrl sys/'<password>'@dg_s2cdb11 <<EOF
convert database s2cdb11 to physical standby
EOF
```

Once the site 2 configuration was completed the database was returned to physical standby mode.

9.2.2. Create a Replica of the Siebel File System from Primary site to Standby Site

A replica of the Siebel File System was created on Site 2. Please see the section Create a Replica of the Siebel File System from Primary site to Standby Site for the additional considerations when creating the file system.

9.2.3. Create Clone of Siebel File System Replica on Standby Site

A clone of the Siebel File System replica was created in the PCA ZFS Storage Appliance for site 2 to facilitate Siebel installation:

PURPOSE	SHARE TYPE	EXPORTED AS
Siebel File System (Clone)	Local	/export/siebelfs_test

Please see the section Site Considerations for the additional considerations when creating the file system.

9.2.4. Site 2 PCA Virtual Machines

Virtual machines were created on Site 2 and configured in the same manner as site 1.

PURPOSE	COUNT OF VMS
Siebel Gateway Servers	3
Siebel Servers	2
Siebel Web Servers	2

9.2.5. Mount the Siebel File System Replica on Siebel Servers

Shared file systems were mounted as follows:

PURPOSE	MOUNTED ON	EXPORTED AS	MOUNTED AS
Siebel File System (Clone)	Siebel Servers	/export/siebelfs_test	/siebelfs

NFS mount options were implemented as follows:

NFS VERSION	MOUNT OPTIONS
v3	_netdev,hard,intr,noatime,rsize=32768,wsize=32768

9.2.6. File System Folders Created

The software folders were created in the same as for site 1.

9.2.7. Database Client Software Installation and Configuration

The 32-bit Oracle database client software was installed on the Siebel Application Servers in the same way as on site 1. Once the software was installed, the Siebel user environment was configured as for site 1.

Connectivity to the database was configured as follows:

```
mkdir –p $ORACLE_HOME/network/admin
cat > $ORACLE_HOME/network/admin/tnsnames.ora <<EOF!
SIEB =
  (DESCRIPTION =
    (CONNECT_TIMEOUT=5)
    (TRANSPORT_CONNECT_TIMEOUT=3)
    (RETRY_COUNT=3)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS = (PROTOCOL = TCP)(HOST = exa14-scan2)(PORT = 1521)))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SIEB))
  )
EOF!
```

9.2.8. Siebel Installation and Configuration

The Siebel installation and configuration were performed in the same way as on site 1.

9.2.9. Shutdown Siebel

Siebel shutdown.

9.2.10. Unmount the Test Siebel File System

The clone Siebel File System was unmounted on each Siebel Server on Site 2.

9.2.11. Return to Physical Standby

The database on site 2 was returned to physical standby mode.

```
dgmgrl sys/'<password>'@dg_s2cdb11 <<EOF
convert database s2cdb11 to physical standby
EOF
```

9.3. Site Test

9.3.1. Physical Standby Database converted to Snapshot Standby

Use Data Guard Broker to convert the standby database to a snapshot standby, for example:

```
dgmgrl sys/'<password>'@dg_s2cdb11 <<EOF
convert database s2cdb11 to snapshot standby
EOF
```

The database service that was configured for snapshot standby mode (SIEB_STBY_TEST) will be started automatically.

9.3.2. Create Clone of Siebel File System Replica on Standby Site

A clone of the Siebel File System replica was created in the PCA ZFS Storage Appliance for site 2 to facilitate Siebel installation:

PURPOSE	SHARE TYPE	EXPORTED AS
Siebel File System (Clone)	Local	/export/siebelfs_test

Please see the section Site Considerations for the additional considerations when creating the file system.

9.3.3. Mount the Siebel File System Replica on Siebel Servers

Shared file systems were mounted as follows:

PURPOSE	MOUNTED ON	EXPORTED AS	MOUNTED AS
Siebel File System (Clone)	Siebel Servers	/export/siebelfs_test	/siebelfs

NFS mount options were implemented as follows:

NFS VERSION	MOUNT OPTIONS
v3	_netdev,hard,intr,noatime,rsize=32768,wsize=32768

9.3.4. Siebel Startup and Test

Start up the Siebel application using the regular process and then Siebel testing was performed.

9.4. Site Test to Standby

9.4.1. Shutdown Siebel

Shut down Siebel on the site test site using the regular process.

9.4.2. Snapshot Standby Database Converted to Physical Standby

Use to Data Guard Broker to convert the snapshot standby database to a physical standby, for example:

```
dgmgrl sys/'<password>'@dg_s2cdb11 <<EOF
convert database s2cdb11 to physical standby
EOF
```

9.4.3. Unmount the Cloned Siebel File System

As root, unmount the cloned Siebel File System used for testing:

```
umount /siebelfs
```

9.4.4. Ready Production Siebel File System Mount Options

So that we are ready for primary operation when necessary, edit the /etc/fstab to point the production Siebel File system on each Siebel Server.

PURPOSE	MOUNTED ON	EXPORTED AS	MOUNTED AS
Siebel File System	Siebel Servers	/export/siebelfs	/siebelfs

NFS mount options were implemented as follows:

NFS VERSION	MOUNT OPTIONS
v3	_netdev,hard,intr,noatime,rsize=32768,wsize=32768

Do not attempt to mount the file system at this time as it will not be available. Note, the mount point (/siebelfs) does not change from Siebel's perspective and so the Siebel configuration does not need to be changed.

9.4.5. Remove the Clone of Siebel File System Replica on Standby Site

PURPOSE	EXPORTED AS
Siebel File System (Clone)	/export/siebelfs_test

9.5. Switchover

The Siebel site switchover can be performed by Oracle Site Guard or manually. The process for using Oracle Site Guard is documented in the technical brief entitled “Application-Level Disaster Recovery using Site Guard”. The manual steps are documented here:

9.5.1. Shutdown Siebel on Primary Site

Shutdown Siebel using the standard procedure and unmount the Siebel File System, for example as root on each Siebel Server and Gateway Server:

```
umount /siebelfs
```

9.5.2. Perform Database Switchover

Use Data Guard Broker to perform the database switchover, for example:

```
dgmgrl sys/'<password>'@dg_s2cdb11 <<EOF
switchover to s2cdb11
EOF
```

9.5.3. Stop Siebel File System Replication at Source

- Locate the Siebel File System project, for example siebelfs.
- Confirm that replication is up-to-date – the “Last Sync” time should be later than when the Siebel File System was dismounted.
- Disable replication and wait for the “STATUS” column to indicate a status of “disabled”.

9.5.4. Perform Siebel File System Role Reversal at Target

- Locate the replica project on the standby (target) site, for example pcasite1:siebelfs.
- Confirm that replication is up-to-date – the “Last Sync” time should be later than when the Siebel File System was dismounted on the old primary site.
- Reverse the direction of replication.
- Enter the new project name, for example “siebelfs”.

9.5.5. Mount the Siebel File System

This procedure should be performed on each Siebel Server.

As root, make sure the current Siebel File System is not mounted:

```
mount /siebelfs
```

9.5.6. Startup Siebel on new primary site

Use the standard procedure to start Siebel.

9.5.7. Start Siebel File System Replication to New Standby Site

A replica of the Siebel File System was created to the new secondary site. Please see the section Create a Replica of the Siebel File System from Primary site to Standby Site for the additional considerations when creating the file system.

9.5.8. Delete Old Siebel File System Project

It is important to delete the old Siebel File System project after the switchover so that a subsequent switchover or failover will not be slowed down by this work.

9.6. Failover

In the event that the primary site is down a failover is performed so that the application can be started on the secondary site. After the failover has been completed it is important to re-establish a standby site as quickly as possible, either by performing the reinstate procedure or by establishing a new secondary site setup.

The Siebel site failover can be performed by Oracle Site Guard or manually. The process for using Oracle Site Guard is documented in the technical brief entitled “Application-Level Disaster Recovery using Site Guard”. The manual steps are documented here:

9.6.1. Perform Database Failover

This activity can be performed in parallel with the Siebel File System Role Reversal. Using Data Guard Broker, the database switchover was performed, for example:

```
dgmgrl sys/'<password>'@dg_2cdb11 <<EOF
failover to s2cdb11
EOF
```

9.6.2. Perform Siebel File System Role Reversal at Target

- Locate the replica project on the standby (target) site, for example pcasite1:siebelfs.
- Confirm that replication is up-to-date – the “Last Sync” time should be later than when the Siebel File System was dismounted on the old primary site.
- Reverse the direction of replication.
- Enter the new project name, for example “siebelfs”.

9.6.3. Mount the Siebel File System

This procedure should be performed on each Siebel Server.

As root, make sure the current Siebel File System is not mounted:

```
mount /siebelfs
```

9.6.4. Startup Siebel as Prod on new primary site

Use the standard procedure to start Siebel.

9.7. Reinstate Standby

9.7.1. Perform database reinstate

Startup one database instance on the new standby (old primary) site, for example:

```
srvctl start instance -d cdb11 -i cdb11
```

Use Data Guard Broker to reinstate the old primary as a physical standby database, for example:

```
dgmgrl sys/'<password>'@dg_2cdb11 <<EOF
reinstate database cdb11
EOF
```

9.7.2. Start Siebel File System Replication to New Standby Site

A replica of the Siebel File System was created to the new secondary site. Please see the section Create a Replica of the Siebel File System from Primary site to Standby Site for the additional considerations when creating the file system.

9.7.3. Delete Old Siebel File System Project

It is important to delete the old Siebel File System project after the reinstating the standby so that a subsequent switchover or failover will not be slowed down by this work.

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com   facebook.com/oracle   twitter.com/oracle

This device has not been authorized as required by the rules of the Federal Communications Commission. This device is not, and may not be, offered for sale or lease, or sold or leased, until authorization is obtained.

Disclaimer: If you are unsure whether your data sheet needs a disclaimer, read the revenue recognition policy. If you have further questions about your content and the disclaimer requirements, e-mail REVREC_US@oracle.com.