Reducing PeopleSoft Downtime Using a Local Standby Database

Oracle Maximum Availability Architecture White Paper
June 2010

Maximum Availability Architecture

Oracle Best Practices For High Availability

ORACLE

Executive Overview 3
Benefits of Using PeopleSoft with a Local Standby Database 5
Oracle Data Guard and Oracle Flashback Database 7
Oracle Data Guard 7
Oracle Flashback Database 8
Configuring Client Failover 8
Setup Procedures 9
Creating a Physical Standby Database 13
Creating a Logical Standby Database 13
Operational Procedures 14
Failing Over to a Local Standby Database 14
Switching Over to a Local Standby Database 15
Upgrading Oracle Database with a Logical Standby Database 16
Upgrading Oracle Database with a Transient Logical Standby 17
Testing with a Physical Standby Database 19
Testing Prerequisites 19
Best Practices 19
PeopleSoft Multi-Pillar Environments 19
The MAA Test Environment 20
Lab Configuration 20
Results 21
Conclusion 21
Appendix A: Manual Procedures 22

Manual Testing with a Physical Standby Database 22
Manually Failing Over to a Local Standby Database 23
Manually Switching Over to a Local Standby Database 24
Appendix B: Client Failover for PeopleSoft 11.1 27
References 32
Oracle MAA White Papers and Demonstrations 32
Oracle Database Documentation 33
My Oracle Support 33

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. MAA white papers are published on the Oracle Technology Network (OTN) at http://www.otn.oracle.com/goto/maa [1].

The PeopleSoft MAA is a best practice blueprint for achieving an optimal PeopleSoft high availability deployment using Oracle high availability technologies and recommendations. This white paper describes setting up PeopleSoft MAA to use Oracle Data Guard standby databases for protection against various outage and maintenance scenarios. Once you have deployed this type of configuration, PeopleSoft applications can fail over to a local standby database to further reduce downtime during planned and unplanned database outages. Figure 1 depicts a PeopleSoft MAA configuration with a local standby database.

Figure 1: PeopleSoft MAA Configured to Use a Local Standby Database

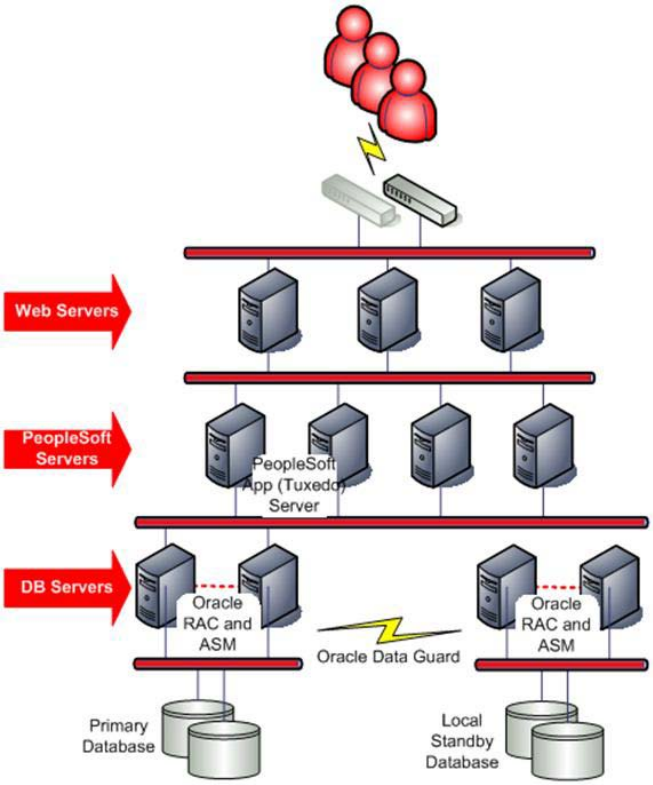
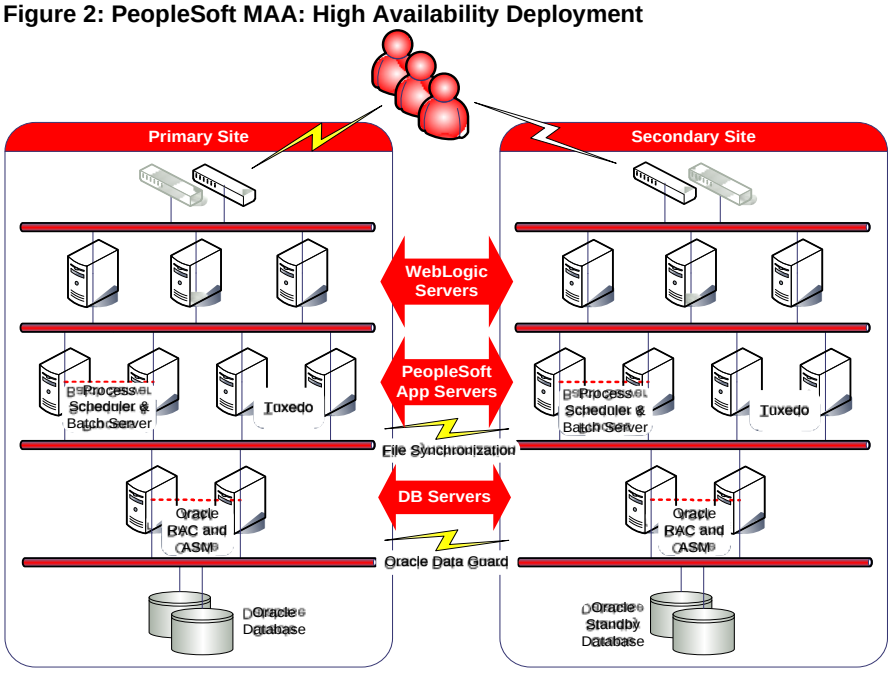


Figure 2 shows a highly available PeopleSoft MAA configuration in which a secondary PeopleSoft standby site has been established for disaster recovery, testing, and other planned maintenance activities. Building a PeopleSoft MAA environment is described in the MAA white paper "Deploying a PeopleSoft Maximum Availability Architecture" [3], which is a companion to this white paper.

Figure 2: PeopleSoft MAA: High Availability Deployment



This white paper is based on PeopleSoft Human Capital Management System, Version 9.1 with PeopleTools Version 8.50 running on Oracle Database 11*g* Release 1 (11.1), and Oracle Database 11*g* Release 2 (11.2). Where necessary, this paper references the documentation for these product releases.

All of the MAA procedures described in this white paper have been tested in our MAA lab. PeopleSoft was configured to stay available during database outages, and to reconnect and resume work when database service resumed. As a result, users experienced only a short pause during the outage and could continue working afterwards. Users did not need to login again, no errors were reported, and no transactions were lost.

Benefits of Using PeopleSoft with a Local Standby Database

Table 1 compares the downtime incurred during unplanned outages when you use a local standby database compared to other solutions.

TABLE 1: USING A LOCAL STANDBY DATABASE TO REDUCE UNPLANNED DOWNTIME

UNPLANNED OUTAGE	ORACLE SOLUTION	DOWNTIME
Database Instance Failure	Oracle RAC Failover	Seconds (transparent)
	Local Standby Failover	Seconds (transparent)
	Remote Site Failover	Minutes to an hour[1]
Corruptions	Active Data Guard (release 11.2)	Zero downtime because auto repair fixes most block corruptions automatically
	Local Standby	Seconds (transparent)
	Remote Site Failure	Minutes to an hour[1]
Entire Database Failure	Local Standby	Seconds (transparent)
	Remote Site Failure	Minutes to an hour[1]

Table 2 compares the downtime incurred during planned maintenance when using a local standby database, compared to other solutions. In addition to the advantages shown in Table 2, another advantage of using the local standby is the ability to validate changes (other than just in your testing environment) prior to failing over all clients and transitioning the standby database to run in the production role.

[1] When you follow the MAA best practices, downtime for database instance failure or entire database failure using remote site failure is measured in minutes to an hour. The key to minimizing the time is automating and integrating database failover with PeopleSoft mid-tier component, as described in the "Deploying a PeopleSoft Maximum Availability Architecture" [3] white paper.

TABLE 2: USING A LOCAL STANDBY DATABASE TO REDUCE PLANNED DOWNTIME

PLANNED MAINTENANCE	ORACLE SOLUTION	DOWNTIME
Database Interim Patch, Database Hardware Upgrade, or Operating System Upgrade	Oracle RAC and Oracle ASM (11*g*) Rolling Upgrade	No downtime
	Local Standby Switchover	Seconds (Transparent)
	Remote Site Switchover	Minutes to an hour[2]
	1. Shutdown 2. Upgrade 3. Startup	Minutes to hours
Database Patch Set or Upgrade	Data Guard database rolling upgrade using the local standby	Seconds (Transparent)
	Remote Site Switchover	Minutes to an hour[2]
	1. Shutdown 2. Upgrade 3. Startup	Hours
Database Transition to Oracle RAC and/or Oracle ASM, Exadata Storage, or Upgrading 10*g* ASM)	Local Standby Switchover	Seconds (Transparent)
	Remote Site Switchover	Minutes to an hour[2]
	1. Shutdown 2. Upgrade 3. Startup	Hours

[2] When you follow the MAA best practices, downtime for database instance failure or entire database failure using remote site failure is measured in minutes to hour. The key to minimizing the time is automating and integrating database failover with PeopleSoft mid-tier components as described in the "Deploying a PeopleSoft Maximum Availability Architecture" [3] paper.

Oracle Data Guard and Oracle Flashback Database

In an environment with a *local* Oracle Data Guard standby database, the PeopleSoft mid-tiers can connect to the primary or standby databases without any noticeable response time impact. Plus, a Data Guard database role transition (where a standby database becomes the new primary database) can be performed without restarting PeopleSoft mid-tiers. Contrast this with a *remote* standby database for which a site failover or a Data Guard role transition requires restarting the PeopleSoft mid-tiers in the secondary site.

Note: It is possible to provide a local standby solution even when the primary and standby databases are located at geographically separate locations, provided the network can handle the load and response time requirements of the application.

Oracle Data Guard

Oracle Data Guard provides the management, monitoring, and automation software infrastructure to create and maintain one or more standby databases to protect Oracle data from failures, disasters, errors, and data corruptions. PeopleSoft supports physical and logical standby databases, Oracle Active Data Guard, and snapshot standby databases.

It is possible to deploy a local standby database at the primary site and a remote standby at the secondary site. The local standby offers the advantage that a failover to the local standby database can be performed automatically with the Data Guard broker fast-start failover, incurring zero data loss while the PeopleSoft servers continue running. The failover can occur almost transparently to the end users. Having both local and remote standby databases also offers the ability to perform a rolling database upgrade to the local standby database. The MAA best practice is to deploy a local standby using Data Guard fast-start failover and a remote standby using the ASYNC log transport mode to achieve maximum availability, data protection, and disaster recovery.

Finally, an important consideration when setting up Data Guard is which of the three modes of data protection you should configure to balance cost, availability, performance, and data protection. Each mode uses a specific redo transport method, and establishes rules that govern the behavior of the Data Guard configuration should the primary database ever lose contact with its standby.

The following table outlines the characteristics of each protection mode.

TABLE 1. DATA GUARD PROTECTION MODES

MODE	RISK OF DATA LOSS	TRANSPORT	IF NO ACKNOWLEDGEMENT FROM THE STANDBY DATABASE, THEN:
Maximum Protection	Zero data loss Double failure protection	SYNC	Stall primary database until acknowledgement is received from the standby database.
Maximum Availability	Zero data loss Single failure protection	SYNC	Stall primary database until acknowledgement is received or `NET_TIMEOUT` threshold period expires. Then resume processing.
Maximum Performance	Potential for minimal data loss	ASYNC	Primary never waits for standby acknowledgment.

See Also:

- *Oracle Data Guard Concepts and Administration* [12] for complete details about Oracle Data Guard
- *Oracle Data Guard Broker* [13] for information about broker management and fast-start failover

Oracle Flashback Database

Oracle Flashback Database enables you to rewind the database to a previous point in time without restoring backup copies of the data files. Flashback Database is a recovery feature that operates on only the changed data. With Flashback Database, the time it takes to correct an error is less than the time it takes to cause and detect the error, without recovery time being a function of the database size. You can flash back a database using a single RMAN command or SQL*Plus statement instead of using a complex procedure. For further details and best practices see the *Oracle Database Backup and Recovery User's Guide* 11*g* Release 1 (11.1) [14] and My Oracle Support (formerly Oracle*Metalink*) Note 565535.1.

Configuring Client Failover

PeopleSoft supports seamless client failover, which allows PeopleSoft to failover database connections to a surviving database or instance when a database connection is lost. You can configure the connections to failover to another Oracle RAC instance, to an Oracle Data Guard standby database, or even to the same database in the case of a database shutdown and restart. The PeopleSoft servers and clients continue running during the failover and do not need to be restarted, and the users need not login again.

The following sections describe how to configure PeopleSoft for client failover for Oracle Database 11*g* release 2 (11.2). For configurations running Oracle Database 11*g* release 1 (11.1), use the instructions in Appendix B.

The configurations require the following:

- Oracle Data Guard broker to manage Data Guard configurations
- Oracle Clusterware for Oracle RAC databases
- Oracle Restart for single-instance databases

Setup Procedures

This section provides step-by-step procedures to configure for client failover and set up PeopleSoft and standby databases to minimize downtime during database outages. The following list presents a high-level description of the steps:

1. Create database services for PeopleSoft database connections and configure the appropriate service attributes.
2. Ensure the client-side Oracle Net configuration points to the database service, not a specific instance, and includes all primary and standby listeners. This ensures that PeopleSoft can connect regardless of where the service is started.
3. Enable Fast Application Notification (FAN) support by applying required patches. For configurations not using FAN, configure Oracle Net timeout parameters.

Create and Configure Services

Using the Server Control (SRVCTL) utility, configure services identically on all databases in the Data Guard configuration. The following example uses SRVCTL commands to create two services: `HCM` and `BATCH` that are enabled for client failover.

```
srvctl add service -d PSFT -s HCM -r "halinux11,halinux12" -P
BASIC –e SELECT –q TRUE –l PRIMARY –z 180 –w 1

srvctl add service -d PSFT -s BATCH -r "halinux11,halinux12" -P
BASIC –e SELECT –q TRUE –l PRIMARY –z 180 –w 1
```

These services have the following attributes:

- Both services are defined on the PSFT database.
- Both services run on Oracle RAC nodes `halinux11` and `halinux12.`
- The client failover policy (`-P`) is `BASIC`
- The client failover type (`-e`) is `SELECT` that allows a `SELECT` statement to fail over following a failure. The application session must not be in an open transaction.

- The client failover High Availability AQ notification (`-q`) is set to `TRUE` for sending FAN events to clients
- The database role in which you want the service to start (`-l`); in the example we only want to start the service if the database is running in the `PRIMARY` role.
- The client failover retries (`-z`) is set to 180 retry attempts
- The client failover retry delay (`-w`) is set to one second

See the SRVCTL ADD SERVICE command in the *Oracle Real Application Clusters Administration and Deployment Guide*.

In addition, database services that are to be active while the database is running in the physical standby role must be created on the primary database using the `DBMS_SERVICE` package. This ensures that information about the service is propagated to the physical standby database so that the services can be started there.

For example:

```
EXECUTE DBMS_SERVICE.CREATE_SERVICE(SERVICE_NAME=>'HCM',
NETWORK_NAME=>'HCM.foo.com', AQ_HA_NOTIFICATIONS=>TRUE,
FAILOVER_METHOD=>'BASIC', FAILOVER_TYPE=>'SELECT',
FAILOVER_RETRIES=>150, FAILOVER_DELAY=>10);
```

Configure the PeopleSoft Application Server

Configuring the PeopleSoft Application Server for seamless client failover consists of properly creating an Oracle Net alias, setting Oracle Net connection attributes, and enabling FAN support.

Step 1 Configure client-side Oracle Net Services.

1. Create an Oracle Net alias that contains all primary and standby listeners in a description list.

```
PSFT=
(DESCRIPTION_LIST =
 (FAILOVER=on)
 (DESCRIPTION =
      (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
       (ADDRESS_LIST=
        (LOAD_BALANCE=on)
(ADDRESS=(PROTOCOL=tcp)(HOST=PRMYSCAN)(PORT=1521))
       (CONNECT_DATA=(SERVICE_NAME=HCM)))
   (DESCRIPTION =
```

```
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
    (ADDRESS_LIST=
     (LOAD_BALANCE=on)
(ADDRESS=(PROTOCOL=tcp)(HOST=STBYSCAN)(PORT=1521))
       (CONNECT_DATA=(SERVICE_NAME=HCM))))
```

When connecting using the above Oracle Net alias, the connection will be routed to the current primary database using the following logic:

a) Client connects using the PSFT Oracle Net alias.

b) Oracle Net contacts DNS and resolves PRMYSCAN to a total of three IP addresses.

c) Oracle Net randomly picks one of the three IP address and attempts to make a connection. If it fails, it tries again for a total of three times.

d) If the connection to primary site is unsuccessful, it then contacts DNS and resolves STBYSCAN to three addresses.

e) It then randomly picks up one of the IP addresses and tries to connect. If it fails, it tries again for a total of three times.

2. When creating the above Oracle Net alias, configure the CONNECT_TIMEOUT and RETRY_COUNT parameters to match your environment:

   a) Set the CONNECT_TIMEOUT parameter to the maximum amount of time (in seconds) to wait for a response from an address before skipping to the next address.

      The timeout interval specified by the CONNECT_TIMEOUT parameter:

      - Is a superset of the TCP connect timeout interval. It includes the time to be connected to the database instance providing the requested service, not just the duration of the TCP connection.
      - Is applicable for each ADDRESS in an ADDRESS_LIST, and each IP address to which a host name is mapped.

      The CONNECT_TIMEOUT parameter is equivalent to the SQLNET.ORA parameter SQLNET.OUTBOUND_CONNECT_TIMEOUT and overrides it.

   b) Set the RETRY_COUNT parameter high enough so that new connection attempts will continue to retry during a failover or switchover operation.

      The RETRY_COUNT parameter specifies the number of times an ADDRESS list is traversed before the connection attempt is terminated.

Also, see the "FastFact: SCAN Overview" for information about the Single Client Access Name (SCAN). SCAN is a new Oracle RAC 11*g* Release 2 feature that provides a single name for clients to access an Oracle Database running in a cluster.

Step 2 Download and apply the patch to enable FAN.

PeopleSoft PeopleTools version 8.50.09 and higher supports FAN. FAN expedites client failover when there is a loss of the primary database. When the primary database is lost and fast-start-failover is initiated, the standby database transitions to the primary database role, and starts the services needed for PeopleSoft to reconnect.

As part of the database service startup, the `AQ_HA_NOTIFICATIONS` parameter causes a FAN event to be sent to all previously connected clients. Upon receipt of the FAN event, the clients break their existing TCP connections and begin failing over by going to the next host in the `TNSNAMES.ORA` connect alias address list until they establish a connection to the new primary database.

To download and apply the required patch, go to My Oracle Support Note 876292.1. PeopleSoft release 8.50.09 is the minimum patch release that contains the FAN functionality.

For more information about the PeopleSoft patch required to accept FAN events, see the "Deploying a PeopleSoft Maximum Availability Architecture" [3] white paper. This paper also includes instructions for configuring the application server to connect to database services.

Notes:

- Some PeopleSoft components are not FAN enabled. This includes Crystal Reports, SQR, some COBOL programs, and third-party components.
- You must restart the PeopleSoft application server for the patch to take effect.

Step 3 Configuring without FAN support

PeopleSoft will support FAN events in a patch release of PeopleTools release 8.50.09 and release 8.51. In releases prior to release 8.50.09, you may need to reduce the value of the TCP Keepalive Timeout parameter for PeopleSoft Application Servers release database connections in the event of a database node crash. This is only for the rare case where the database node crashes before the TCP connections can be cleaned up, and only for connections where a database request was in-flight at the time of failure or a new request was started before the Virtual Internet Protocol (VIP) Address could be switched to a surviving node. In all other cases, the database connection failure is detected and a new connection is established on a surviving node.

You can also control TCP timeout at the Oracle Net alias level. The `TRANSPORT_CONNECT_TIMEOUT` parameter specifies the time, in seconds, for a client to establish a TCP connection to the database server. The default value is 60 seconds. The

timeout interval is applicable for each `ADDRESS` in an `ADDRESS_LIST` description, and each IP address that a host name is mapped. The `TRANSPORT_CONNECT_TIMEOUT` parameter is equivalent to the sqlnet.ora parameter `TCP.CONNECT_TIMEOUT`, and overrides it.

Because these configuration changes may have adverse effects on network utilization, you should carefully test and monitor all changes.

Creating a Physical Standby Database

The process of creating a physical standby database is documented in the "Creating a Physical Standby Database" section of *Oracle Data Guard Concepts and Administration*. There are no special steps for creating a physical standby database for use as a PeopleSoft database. For Oracle RAC databases, see the "Data Guard 11*g* Installation and Configuration on Oracle RAC Systems" white paper for MAA best practices at http://www.oracle.com/technology/deploy/availability/pdf/dataguard11g_rac_maa.pdf.

Note:

- Ensure archiving is enabled.

  If archiving is not enabled on your database, then you must shut down and restart the database to enable archiving.

- Ensure Flashback Database is enabled.

  For fast-start failover, you must enable Flashback Database on both the primary and physical standby databases.

Creating a Logical Standby Database

The process of creating a logical standby database is documented in the "Creating a Logical Standby Database" section of *Oracle Data Guard Concepts and Administration* [12]. For Oracle RAC configurations, see the "MAA / Data Guard 10g Release 2 Setup Guide – Creating a RAC Logical Standby for a RAC Primary" white paper at http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_RACPrimaryRACLogicalStandby.pdf.

> **Note:** The out-of-the-box PeopleSoft database uses only supported types and attributes that a logical standby database maintains. However, you must check any additional objects that have been created in the database. To check for supported data types and storage attributes for the database, see Appendix C of the *Oracle Data Guard Concepts and Administration* documentation.

Operational Procedures

This section describes the operational procedures associated with various PeopleSoft database outage scenarios.

Failing Over to a Local Standby Database

Failing over to a local standby database is typically performed if the primary database service is lost. There are a number of scenarios where this may happen. For example, if the database crashes and cannot be restarted, or if the database files have become corrupt. Having the facility to failover to a standby database typically saves many hours of downtime.

Failover Prerequisites

The failover procedures described in this section assume:

- A local standby database is in place.

  Having a local standby database allows the failover to be performed transparently to the PeopleSoft users.

- PeopleSoft is up and running against the primary database.
- A mechanism is in place to alert administrators in the event of a database failure so that they can respond accordingly.
- If fast-start failover has been implemented, then the database recovery is initiated automatically.

Failover Best Practices

- Automatically fail over PeopleSoft applications to a local standby database using Data Guard fast-start failover
- Configure Data Guard with the Maximum Availability data protection mode. With this configuration, the database will fail over and incur zero data loss, and users can continue seamlessly. If data loss is incurred, then manual intervention is required and users have to be alerted.

  Note: If fast-start-failover management by the Data Guard broker is **not** enabled, then you must perform the manual actions described in the "Manually Switching Over to a Local Standby Database" section of this white paper.

- Follow the steps in "Fast-Start Failover Best Practices: Oracle Data Guard 10g Release 2" white paper to configure fast-start failover. The procedures in the white paper are also applicable to Oracle Database 11*g* Release 1 and 11g Release 2.

- For more best practices, see:
  - "Oracle Data Guard Switchover and Failover Best Practices" MAA white paper at http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_SwitchoverFailoverBestPractices.pdf
  - *Oracle Database High Availability Best Practices* documentation at http://www.oracle.com/pls/db111/lookup?id=HABPT

Switching Over to a Local Standby Database

Switchover to a local standby database may become necessary for planned maintenance activities, such as to perform a hardware upgrade, to transition to Oracle RAC or Oracle ASM, or in the case where the primary database environment is hung or seriously underperforming due to hardware or system issues.

Switchover Prerequisites

The following switchover procedures assume:

- A local standby database is in place. Having a local standby allows the switchover to be performed transparently to the PeopleSoft users.
- PeopleSoft is up and running against the primary database.

Switchover Best Practices

- Use the ability to switch over to a standby database to save many hours of downtime during planned maintenance scenarios.
- Use the Data Guard broker and Enterprise Manager Grid Control to perform the switchover.
- Use a local standby database for the switchover, if possible, so that the switchover occurs transparently to PeopleSoft users.

Notes:

- If you do not use the Data Guard broker and Enterprise Manager Grid Control to perform switchover, then you must follow the manual procedures in the "Manually Switching Over to a Local Standby Database" section of this white paper..
- During a switchover, all committed transactions on the primary database are recovered on the standby database and no data is lost.
- Switch back is always possible.
- The procedure to switchover during a database upgrade is different from a regular switchover. See the "Upgrading Oracle Database with a Local Logical Standby" section for details.

For additional best practices, see:

- "Oracle Data Guard Switchover and Failover Best Practices" MAA white paper at http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_SwitchoverFailoverBestPractices.pdf
- *Oracle Database High Availability Best Practices* documentation at http://www.oracle.com/pls/db111/lookup?id=HABPT

Upgrading Oracle Database with a Logical Standby Database

This section describes how to perform a rolling database upgrade using an existing logical standby database while the PeopleSoft application is up and running.

> **Note:** If your Data Guard configuration only includes physical standby databases, perform the upgrade using a transient logical standby database as described in the "Upgrading Oracle Database with a Transient Logical Standby Database" section.

In this scenario:

- A database version upgrade is performed while PeopleSoft continues running.
- There is a short pause during the switchover to the upgraded database, so you should warn the end users to expect a brief pause in service.
- All committed transactions on the primary database are recovered on the standby and no data is lost.
- Once you are satisfied with the performance on the newly upgraded system, you can upgrade the standby database and resume Redo Apply. If the transient logical procedures were utilized, then follow the steps described in the "Upgrading Oracle Database Using a Transient Logical Standby" section to reinstate the physical standby database
- The two-phased commit (prepare and then commit) procedure is not supported for database upgrade switchovers.
- The Data Guard broker does not support the rolling upgrade process, so you must disable the broker for the duration of the upgrade.

Perform the following steps to complete the upgrade:

1. Establish a local logical standby database for the primary PeopleSoft database and allow SQL Apply to bring the standby database up-to-date. The local logical standby can be created as a new database, or by temporarily converting a physical standby database to run in the logical standby role.
2. Stop SQL Apply on the logical standby:

```
SQL> ALTER DATABASE STOP LOGICAL STANDBY APPLY;
```

3. Upgrade the logical standby using the standard database upgrade procedure described in the *Oracle Database Upgrade Guide*.

4. Re-start SQL Apply and bring the standby database up-to-date:

   ```
   SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
   ```

   Note: It is possible to test the newly upgraded database at this point before the switchover.

5. On the primary database, stop the database service, disconnect all the PeopleSoft connections, and commit to the standby database role:

   ```
   SQL> EXECUTE DBMS_SERVICE.STOP_SERVICE('HCM');
   SQL> EXECUTE DBMS_SERVICE.DISCONNECT_SESSION('HCM');
   SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO LOGICAL STANDBY;
   ```

   At this point, access for the PeopleSoft end users will pause when submitting work and must wait until the switchover has completed.

6. On the standby database, query the V$DATABASE view until the switchover status becomes `'TO PRIMARY'`:

   ```
   SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
   ```

7. Switch the standby database to the primary role:

   ```
   SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;
   ```

   When the database changes to the primary role, the role change trigger starts the PeopleSoft database service. Transparent client failover establishes connections to the new primary database and PeopleSoft resumes working.

Upgrading Oracle Database with a Transient Logical Standby

The "Upgrading Oracle Database Transparently with a Logical Standby Database" section described how to perform a rolling database upgrade using an existing logical standby database while the PeopleSoft application is up and running. However, if your configuration only includes physical standby databases, you should consider performing the upgrade using a *transient logical standby database*

A transient logical standby is an existing physical standby database that has been converted temporarily to a logical standby for the purpose of performing the rolling database upgrade. The actual upgrade is performed on the logical standby database, after which it assumes the PRIMARY role while the original primary database is upgraded through Redo Apply. Once both the logical standby and primary databases upgrade is complete, the logical standby database can be returned to its original physical standby role.

There are several benefits for using a transient logical standby database:

- No new standby database needs to be instantiated, an existing physical standby is used.
- The physical standby database is converted to a logical standby using the `KEEP IDENTITY` parameter that was introduced in Oracle Database 11*g* Release 1. This parameter preserves the `DBNAME` and `DBID`.
- You only need to perform an upgrade once (using Oracle Universal Installer or manually).
- PeopleSoft is up and running throughout the upgrade process, although users may experience a pause during the switchover part of the process.

The procedures for performing a rolling upgrade are detailed in the MAA white paper: "Rolling Database Upgrades for Physical Standby Databases using Transient Logical Standby 11g."

Upgrade Prerequisites

To use this procedure, the following requirements must be met:

- Ensure the starting Oracle Database release running on the primary and physical standby databases is Oracle Database 11*g* Release 1 (11.1.0.7).
- Enable Flashback Database for the primary and physical standby databases. In addition, you must create guarantee restore points.
- Define a fast recovery area on the primary and physical standby databases.
- Disable Data Guard broker management, including disabling fast-start failover.

Upgrade Best Practices

- Use the automated method to perform a database rolling upgrade using a transient logical standby. See the MAA white paper describing the automation: "Database Rolling Upgrades Made Easy." The current version of the automation scripts only support non-Oracle RAC environments. For Oracle RAC environments, use the step-by-step process documented in the "Rolling Database Upgrades for Physical Standby Databases using Transient Logical Standby 11g." MAA white paper.
- As with all procedures, Oracle recommends perfecting and automating the operational procedures and upgrade procedures on a test environment before performing the database upgrade on production systems.
- Keep the original setting of the `COMPATIBLE` parameter until you have completed the rolling upgrade and have addressed any issues for PeopleSoft on the upgrade database.

Testing with a Physical Standby Database

You can use a local and remote physical standby database to test the effect of the failover, switchover, and upgrade procedures described in this white paper while the primary database is in live operation. When you are satisfied with the functionality on the test system, you can roll it out to the production database, and then use Oracle Flashback Database to quickly restore the database to standby operation afterwards. This technique can also be used during a transparent database upgrade to test the new database version prior to the switchover.

Testing Prerequisites

In this testing scenario, you must meet the following pre-requisites:

- A physical standby database is in place with Flashback Database enabled.
- PeopleSoft is up and running against the primary database.

Best Practices

- Use the broker's Oracle Enterprise Manager Grid Control interfaced to perform centralized, one-click testing.

  From the Central Console of Enterprise Manager Grid Control, you can perform all testing and management operations either locally or remotely.

- Use a snapshot standby database

  Beginning with Oracle Database 11*g* Release 1, Oracle Data Guard provided a snapshot standby database as a way to use a physical standby database for testing. When the physical standby is converted to a snapshot standby, redo information is still transported to the standby database but the redo data is not applied. This method allows you to use the standby database for testing purposes but it still preserves recovery point objective (RPO).

The manual procedure for performing testing with a physical standby database is documented in Appendix A.

PeopleSoft Multi-Pillar Environments

When multiple PeopleSoft Suites are deployed, they each require their own database and application and web server tier components. For example, if your enterprise is running PeopleSoft Human Capital Management, Financials Management, and Customer Relation Management, then you would have three separate databases and application servers or three pillars. These pillars are for the most part, independent of each other and should be treated as such.

For multi-pillar implementations, implement the following MAA guidelines:

1. For each pillar, apply the same MAA best practices as we have done for HCM described above.
2. Each pillar will require its own Data Guard Broker configuration and its own Fast-Start Failover observer
3. Perform failover and switchover testing of each pillar to ensure these processes are functioning properly.
4. In a local standby database configuration, you may switchover or failover PeopleSoft pillars independently and in any order because each pillar's services will be maintained by RAC or the local standby database provided you follow the above MAA best practices.

The MAA Test Environment

The operational procedures described in this paper were tested under the following conditions

- The database upgrade was performed from Oracle Database 11*g* releases 11.1.0.7 to 11.2.0.1. All other procedures were performed on Oracle Database 11*g* release 11.1.0.7.
- All tests were performed using an Oracle RAC primary and a single instance (non-Oracle RAC) databases.
- All databases were configured with Oracle Automatic Storage Management (Oracle ASM).
- All procedures were performed on PeopleSoft PeopleTools 8.50.01 with HCM 9.1.

Lab Configuration

The tests were developed and executed using HP Systems hardware and software.

HP Hardware

- HP Proliant DL385 G2 running PeopleSoft application and Web servers
- HP Proliant DL385 G2 servers (2) for the 2-node RAC database tier
- HP Proliant DL385 G2 servers for the standby database tier
- HP StorageWorks EVA4000 for database storage

Operating System

- Redhat Enterprise Linux 5.2 (64bit) for Web and PeopleSoft Servers
- Redhat Enterprise Linux 5.2 (64bit) for Database Servers

Results

In all tests, the PeopleSoft server processes continued running during the database outage and reconnected and resumed work when the database transition was completed.

Conclusion

Businesses with mission-critical, enterprise wide deployments need solutions that reduce downtime, protect against data losses, and provide higher utilization of their infrastructure. Using a local standby database and following the MAA best practices yields the highest availability for local planned and unplanned outages, protects against data loss, and provides allows you to use the local standby database for multiple purposes, including real-time queries (Oracle Active Data Guard), application testing, and validation using the Data Guard snapshot standby feature.

Oracle Database 11*g* Release 2 (11.2) coupled with the Oracle Active Data Guard option provides transparent block corruption repair. These benefits reduce total cost of ownership by leveraging existing infrastructure for these purposes. You can achieve all of these benefits without having to shut down the PeopleSoft applications. Using the fast-start failover feature and Fast Application Notification (FAN), the user community will experience minimal (or no) impact.

Appendix A: Manual Procedures

- Manual Testing with a Physical Standby Database
- Manually Failing Over to a Local Standby Database
- Manually Switching Over to a Local Standby Database

Manual Testing with a Physical Standby Database

The recommended method for performing testing is described in the "Testing with a Physical Standby Database" section. To perform manual testing, perform the following events on the standby database:

1. Cancel Redo Apply on the standby database:

   ```
   SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
   ```

2. Convert the standby to a snapshot standby and then restart the database:

   ```
   SQL> ALTER DATABASE CONVERT TO SNAPSHOT STANDBY;
   SQL> SHUTDOWN IMMEDIATE
   SQL> STARTUP;
   ```

   Note: If you use a database trigger, then the trigger will not fire because the database role will be snapshot standby. Also, if your standby database is an Oracle RAC database, you must shutdown all but one instance to perform the conversion to a snapshot standby. Once the conversion to a snapshot standby is complete, you may start up all remaining Oracle RAC instances.

3. Use the snapshot standby database for testing.

   The database is now open and can be used for testing. Any changes that are made to the database will be rolled back afterwards when the database is converted back to a physical standby. You can start additional databases instances and test the application, as necessary. Remember that during the testing period redo data is not applied to the snapshot standby database and it will lag behind the primary database. For availability and fast recoverability, consider using another physical standby.

4. When you have finished testing on the snapshot standby, shut down all but one database instance (if running Oracle RAC).

5. Convert the database back to a physical standby, restart the database, and start Redo Apply:

   ```
   SQL> STARTUP MOUNT FORCE;
   SQL> ALTER DATABASE CONVERT TO PHYSICAL STANDBY;
   ```

```
SQL> STARTUP MOUNT FORCE;
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT
LOGFILE DISCONNECT;
```

6. Redo apply on the physical standby will apply all outstanding redo data to bring the physical standby up to date with the primary database.

Manually Failing Over to a Local Standby Database

If fast-start-failover management by the Data Guard broker is *not* enabled, then you must perform the manual actions described in the following sections to complete the recovery.

Failing Over to a Local Physical Standby Database

When the primary database fails, all PeopleSoft database connections are lost and begin cycling through the address list trying to reconnect. :

1. Begin failing over to the standby database by executing the following command on the standby database:

   ```
   SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE FINISH;
   SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;
   SQL> ALTER DATABASE OPEN;
   ```

   When the failover has completed, the standby database becomes the primary and the database role change trigger starts the database service. A FAN event is sent to all previously connected sessions, client failover establishes connections to the new primary database and PeopleSoft resumes working.

   Note: If the standby database was ever opened read-only then it must be restarted prior to opening. For additional best practices, see the "Oracle Data Guard Switchover and Failover Best Practices" MAA white paper at http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_SwitchoverFailoverBestPractices.pdf

2. When the original primary database becomes available, it can be reinstated as a standby for the new primary database.

   This can be done either by restoring a backup or by using Oracle Flashback Database.

   > **Note:** The steps are documented in *Oracle Data Guard Concepts and Administration*. If Oracle Flashback Database is enabled on the failed primary database, see Section 13.2.1 "Flashing Back a Failed Primary Database into a Physical Standby Database" and follow the steps for reinstating the old primary database to be the new physical standby database.

Failing Over to a Local Logical Standby Database

When the primary database fails, all PeopleSoft database connections are lost and begin cycling through the address list trying to reconnect. Perform the following actions to complete the recovery:

1. Begin failing over to the standby database by executing the following command on the standby:

   ```
   SQL> ALTER DATABASE ACTIVATE LOGICAL STANDBY DATABASE
   FINISH APPLY;
   ```

   When the failover has completed the standby becomes the primary, the role change trigger starts the database service, client failover works to establish connections to the new primary database, and PeopleSoft resumes working.

2. When the original primary database becomes available, it can be reinstated as a standby for the new primary database.

This can be done either by restoring a backup or by using Flashback Database. The steps are documented in *Oracle Data Guard Concepts and Administration*. If Flashback Database is enabled on the failed primary database, see Section 13.2.2 "Flashing Back a Failed Primary Database into a Logical Standby Database" and follow the steps for reinstating the failed primary into a new logical standby database.

Manually Switching Over to a Local Standby Database

The recommended method for switching over to a local standby database is described in the "" section of this white paper. If management by the Data Guard broker is *not* enabled, then you must perform the manual actions described in the following sections.

Switching Over to a Local Physical Standby Database

All switchover operations can be done through Enterprise Manager or with one Data Guard Broker command. The manual steps below are described.

1. Run the following command to switchover the primary database to the standby role:

   ```
   SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO STANDBY WITH SESSION
   SHUTDOWN;
   ```

   All PeopleSoft connections are lost and begin cycling through the address list to reconnect PeopleSoft users.

2. Shutdown the primary database and restart it in mount mode:

```
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP MOUNT
```

3. Query the V$DATABASE view to check the switchover status.

   On the standby database, issue the following query until the switchover status becomes either 'TO PRIMARY' or 'SESSIONS ACTIVE':

```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
```

   Then, commit the standby to switch over to the primary role and open the database:

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;
SQL> ALTER DATABASE OPEN;
```

   When the database opens in the primary role, the startup trigger starts the PeopleSoft database service. A FAN event is sent to the previously connected sessions, clients fail over connections to the new primary database and PeopleSoft resumes working.

   Note: If the standby database was ever opened read-only, then it must be restarted prior to opening for normal use. For more details see the "Switchover and Failover Best Practices" white paper at http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_SwitchoverFailoverBestPractices.pdf

   The original primary database now becomes the standby database.

4. Start Redo Apply on the new standby:

```
SQL> RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE
DISCONNECT;
```

5. Switch logs on the new primary database:

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
```

Switching Over to a Local Logical Standby Database

All switchover operations can be done through Oracle Enterprise Manager or with one Data Guard broker command. Furthermore, if you have a physical standby database, you can temporarily convert the physical standby database to a logical standby through the transient logical standby procedures. See the "Upgrading Oracle Database Using a Transient Logical Standby" section in this white paper.

Use the following steps to perform a manual switchover:

1. Prepare the primary database to switch over to the standby role by running the following command on the primary:

   ```
   SQL> ALTER DATABASE PREPARE TO SWITCHOVER TO LOGICAL STANDBY;
   ```

2. Prepare the standby database to switch over to the primary role by running the following command on the standby:

   ```
   SQL> ALTER DATABASE PREPARE TO SWITCHOVER TO PRIMARY;
   ```

3. Query the V$DATABASE view and wait until the switchover status on the primary changes to 'TO LOGICAL STANDBY':

   ```
   SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
   ```

   Note: PeopleSoft continues running normally up to this point.

4. On the primary database, stop the database service, disconnect all the PeopleSoft connections, and commit to the standby role:

   ```
   SQL> EXECUTE DBMS_SERVICE.STOP_SERVICE('HCM');
   SQL> EXECUTE DBMS_SERVICE.DISCONNECT_SESSION('HCM');
   SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO LOGICAL STANDBY;
   ```

5. On the standby database, query the V$DATABASE view until the switchover status becomes 'TO PRIMARY':

   ```
   SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
   ```

   Then, commit the standby to switch over to the primary role:

   ```
   SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;
   ```

   When the database changes to the primary role, the role change trigger starts the PeopleSoft database service. Clients fail over by establishing connections to the new primary database and PeopleSoft resumes working.

   Note: When the logical standby becomes the primary, no FAN events are sent to the clients running Oracle Database 11*g* Release 1. Client failover continues to work. Our MAA testing shows that PeopleSoft reconnected to the new primary database with no restart of the application server required. FAN events will occur in Oracle Database 11*g* Release 2 (11.2). The role base trigger fires to start the services on the new primary.

6. The original primary is now the standby database. Begin SQL Apply on the new standby:

   ```
   SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
   ```

   Switch logs on the new primary: `SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;`

Appendix B: Client Failover for PeopleSoft 11.1

PeopleSoft supports seamless client failover, which allows PeopleSoft to failover database connections to a surviving database or instance when a database connection is lost. You can configure the connections to failover to another Oracle RAC instance, to an Oracle Data Guard standby database, or even to the same database in the case of a database shutdown and restart. The PeopleSoft servers and clients continue running during the failover and do not need to be restarted, and the users need not login again.

The following steps illustrate how to configure PeopleSoft for client failover for Oracle Database 11*g* release 1 (11.1).

Perform the following steps:

Step 1 For Oracle RAC environments only, configure Oracle Clusterware Managed Database Services.

Note: Perform this step only if your environment is Oracle RAC. Otherwise, skip to step 3.

Create an Oracle Clusterware managed database service for PeopleSoft connections to the database. This is necessary to ensure that connections are made only to open database instances.

Use the following instructions if you are using Oracle grid infrastructure 11*g* release 1 (11.1), as appropriate.

Oracle Clusterware managed database services are created through Oracle Enterprise Manager or by using the `SRVCTL` command.

The following example adds a service using the `SRVCTL` command:

```
srvctl add service -d PSFT -s HCM -r "halinux11,halinux12" –P BASIC
srvctl add service -d PSFT -s BATCH -r "halinux11,halinux12" -P BASIC
```

The example creates two services, `HCM` and `BATCH`, which are enabled for client failover with the following attributes:

- Both services are defined on the PSFT database.
- Both services run on Oracle RAC nodes `halinux11` and `halinux12.`
- The client failover policy (`-P`) is `BASIC`

These services are added into the Oracle Cluster Repository (OCR) and defined in the PSFT database. You do need to use the `DBMS_SERVICE` PL/SQL package to create the service within the database.

Step 2 For Oracle RAC environments only, modify the service to enable high-availability notification to be sent through Advanced Queuing (AQ).

Note: Perform this step only if your environment is Oracle RAC. Otherwise, skip to step 3.

If you have created the database service as instructed in step 1, then use the DBMS_SERVICE package to modify the service to enable high-availability notification to be sent through Advanced Queuing (AQ) by setting the AQ_HA_NOTIFICATIONS attribute to TRUE. To configure server-side settings for client failover, set the FAILOVER attributes, as shown in the following example:

```
exec DBMS_SERVICE.MODIFY_SERVICE(
service_name => 'HCM',
   aq_ha_notifications => true,
   failover_method => 'BASIC',
   failover_type => 'SELECT',
   failover_retries => 180,
   failover_delay => 1);
```

Note: See the *Oracle Database PL/SQL Packages and Types Reference* for more information about the DBMS_SERVICE package.

Step 3 Create the database service and enable high-availability notification, and configure server-side client-failover settings.

If you are creating database services for a single-instance (non-Oracle RAC) database, or if you are creating services for an Oracle RAC database and are not using Enterprise Manager, use the CREATE_SERVICE subprogram to both create the database service and enable high-availability notification and configure server-side settings for client failover. To enable Fast Application Notification and adjust the configuration for each service, execute the following SQL*Plus statements while connected as the "SYS" user on the primary::

```
exec DBMS_SERVICE.CREATE_SERVICE (
 service_name => 'HCM',
 network_name => 'HCM',
       aq_ha_notifications => true,
       failover_method => 'BASIC',
       failover_type => 'SELECT',
       failover_retries => 180,
       failover_delay => 1);

exec DBMS_SERVICE.CREATE_SERVICE (
 service_name => 'BATCH',
 network_name => 'BATCH',
```

```
aq_ha_notifications => true,
failover_method => 'BASIC',
failover_type => 'SELECT',
failover_retries => 180,
failover_delay => 1);
```

Step 4 Create a trigger that fires on the system startup event to relocate the database service.

Create a trigger that fires on the system startup event to relocate the database service 'HCM' to a Data Guard standby database (Oracle RAC or non-Oracle RAC) after it has transitioned to the primary role.

```
CREATE OR REPLACE TRIGGER manage_OCIservice
after startup on database
DECLARE
   role VARCHAR(30);
BEGIN
   SELECT DATABASE_ROLE INTO role FROM V$DATABASE;
   IF role = 'PRIMARY' THEN
   DBMS_SERVICE.START_SERVICE('HCM');
END IF;
   END;
```

If the Data Guard configuration is not using real-time apply, then archive the current redo log file to be sure that the changes are applied to the standby database:

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
```

Step 5 Configure Oracle Net (TNSNAMES.ORA file) to use services.

After creating the services using the Oracle Clusterware SVRCTL utility, define Oracle Net aliases in the TNSNAMES.ORA file to reference the services. The following example uses two TNS aliases to refer to the HCM and BATCH services:

```
PSFT =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS=(PROTOCOL=TCP)(HOST=halinux11-vip)(PORT=1521))
      (ADDRESS=(PROTOCOL=TCP)(HOST=halinux12-vip)(PORT=1521))
    )
    (LOAD_BALANCE=on)
```

```
    (CONNECT_DATA=
      (SERVER=DEDICATED)
      (SERVICE_NAME=HCM)
      )
    )
  )

BATCH =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS=(PROTOCOL=TCP)(HOST=halinux11-vip)(PORT=1521))
      (ADDRESS=(PROTOCOL=TCP)(HOST=halinux12-vip)(PORT=1521))
    )
    (LOAD_BALANCE=on)
    (CONNECT_DATA=
      (SERVER=DEDICATED)
      (SERVICE_NAME=BATCH)
      )
    )
  )
```

When configuring the PeopleSoft Application Servers, specify the `DBNAME` within the `psadmin` configuration utility to be the TNS alias `PSFT`, the `PEOPLESOFT` domain, and "`BATCH`" for the process scheduler in the same `PEOPLESOFT` domain. The application server and process scheduler will now inherit all of the specified client failover attributes. See Appendix A for detailed steps to configure the PeopleSoft Application Server.

Step 6 Set `SQLNET.OUTBOUND_CONNECT_TIMEOUT` parameter in `SQLNET.ORA`.

When connecting or reconnecting to the database, PeopleSoft will keep trying all the addresses in the address list until a connection is established. It is important that it does not become stalled on any one address, especially one from an inaccessible server node, because this will delay the connection.

Set the `SQLNET.OUTBOUND_CONNECT_TIMEOUT` parameter to the maximum amount of time (in seconds) to wait for a response from an address before skipping to the next address. A minimum setting of three seconds is recommended and is acceptable in most cases. For example:

```
SQLNET.OUTBOUND_CONNECT_TIMEOUT=3
```

Make the configuration changes for all PeopleSoft clients and servers that connect directly to the database. This parameter is set in the `SQLNET.ORA` file.

Step 7 Apply the PeopleSoft Application Server patch set.

PeopleSoft PeopleTools version 8.50.09 (and later releases) support Fast Application Notification (FAN). FAN expedites client failover when there is a loss of the primary database or when an Oracle RAC experiences node or instance failure.

To enable FAN, perform the following tasks:

- Patch the application server component of PeopleSoft to enable its OCI-based client to receive FAN events.

  The minimum patch number that contains the FAN functionality is release 8.50.09. Any maintenance patch at or higher than this version supports the FAN functionality. To download and apply the required patch, go to My Oracle Support Note 876292.1.

- Apply the patch for the Oracle Database client releases 9.2.0.8, 10.2.0.3 and 11.1.0.7. Apply this patch to the `ORACLE_HOME` client to which the PeopleTools installation points. See My Oracle Support Note: 1091386.1 "PeopleSoft Queries with Transparent Application Failover (TAF)".

  Note: Oracle Data Guard and Oracle RAC both support FAN events. FAN events fire when:

  - The primary database is lost and fast-start-failover or any broker-managed Data Guard failover is initiated.

    The standby database transitions to run in the new primary database role and the database startup trigger fires to start the services needed for PeopleSoft to reconnect. As part of the transition from standby to primary, the `AQ_HA_NOTIFICATIONS` property of a service (that is enabled for client failover) that is set to `TRUE` causes a FAN event to be sent to all previously connected clients.

  - An Oracle RAC node or an Oracle RAC instance fails.

    The clusterware detects the failure and a surviving Oracle RAC instance notifies the affected clients.

  Upon receipt of the FAN event, the clients break the existing TCP connections and initiate client failover, which references the `TNSNAMES.ORA` connect alias address list and establishes a connection to the surviving instance.

Step 8 Configure the TCP Keepalive Timeout parameter.

Note: Perform this step only if you *did not apply* the PeopleTools patch set for release 8.50.09 and later releases.

For releases prior to release 8.50.09, you may need to reduce the value of the TCP Keepalive Timeout parameter for PeopleSoft Application Servers to release database

connections in the event of a database node crash. This is only for the rare case where the database node crashes before the TCP connections can be cleaned up, and only for connections where a database request was in-flight at the time of failure or a new request was started before the Virtual Internet Protocol (VIP) Address could be switched to a surviving node. In all other cases, the database connection failure is detected quickly and a new connection is established on a surviving node.

See the My Oracle Support Note 249213.1 - Performance problems with Failover when TCP Network goes down (no IP address) to configure the TCP Keepalive timeout.

Note: Changing the TCP Keepalive timeout parameter may have adverse effects on other network users.

Step 9 Restart the mid-tiers automatically.

If this is remote standby with its own set of mid-tiers, you can configure triggers based on the `DB_ROLE_CHANGE` system event to startup your mid-tiers. Configuring triggers is described in the "Deploying a PeopleSoft Maximum Availability Architecture" [13] white paper.

References

Oracle MAA White Papers and Demonstrations

1. Oracle Maximum Availability Architecture Web site
   http://www.otn.oracle.com/goto/maa
2. MAA demonstrations page:
   http://www.oracle.com/technology/deploy/availability/demonstrations.html
3. "Deploying a PeopleSoft Maximum Availability Architecture" white paper.
   http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm
4. "Creating an Oracle RAC Physical Standby for an Oracle RAC Primary" white paper
   http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10g_RACPrimaryRACPhysicalStandby.pdf
5. "Creating an Oracle Logical Standby for an Oracle Primary" white paper
   http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_RACPrimaryRACLogicalStandby.pdf
6. "Best Practices: Data Guard Fast-Start Failover" white paper
   http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_FastStartFailoverBestPractices.pdf

7. "Database Rolling Upgrade With Transient Logical Standby" white paper http://www.oracle.com/technology/deploy/availability/pdf/maa_wp_11g_transientlogicalrollingupgrade.pdf

8. Client Failover in Data Guard Configurations for Highly Available Oracle Databases http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_ClientFailoverBestPractices.pdf

Oracle Database Documentation

9. *Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration* http://download.oracle.com/docs/cd/E15645_01/pt850pbr0/eng/psbooks/tsvt/book.htm?File=tsvt/htm/tsvt14.htm#H4003

10. *Oracle Database High Availability Overview* http://www.oracle.com/pls/db111/lookup?id=HAOVW

11. *Oracle Database High Availability Best Practices* http://www.oracle.com/pls/db111/lookup?id=HABPT

12. *Oracle Data Guard Concepts and Administration* http://www.oracle.com/pls/db111/lookup?id=SBYDB

13. *Oracle Data Guard Broker* http://www.oracle.com/pls/db111/lookup?id=DGBKR

14. *Oracle Database Backup and Recovery User's Guide* ("Rewinding a Database with Flashback Database") http://www.oracle.com/pls/db111/lookup?id=BRADV89367

My Oracle Support

15. Note 460982.1 - How To Configure Server Side Transparent Application Failover

16. Note 453293.1 - 10g: Configuration of TAF (Transparent Application Failover) and Load Balancing

17. Note 249213.1 - Performance problems with Failover when TCP Network goes down (no IP address)

18. Note 565535.1 - Flashback Database Best Practices & Performance

Reducing PeopleSoft Downtime Using a Local Standby Database
June 2010
Author: Darryl Presley
Contributing Authors: Ravi Shankar, Suhas Kulkarni, Jerry Zarate, Linda Koose, Bernice Eng, Lawrence To, Mike Smith, Nancy Ikeda
Viv Schupmann (Editor)

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment

0109