Deploying an Oracle PeopleSoft Maximum Availability Architecture

Oracle Maximum Availability Architecture White Paper
February 2011

Maximum Availability Architecture

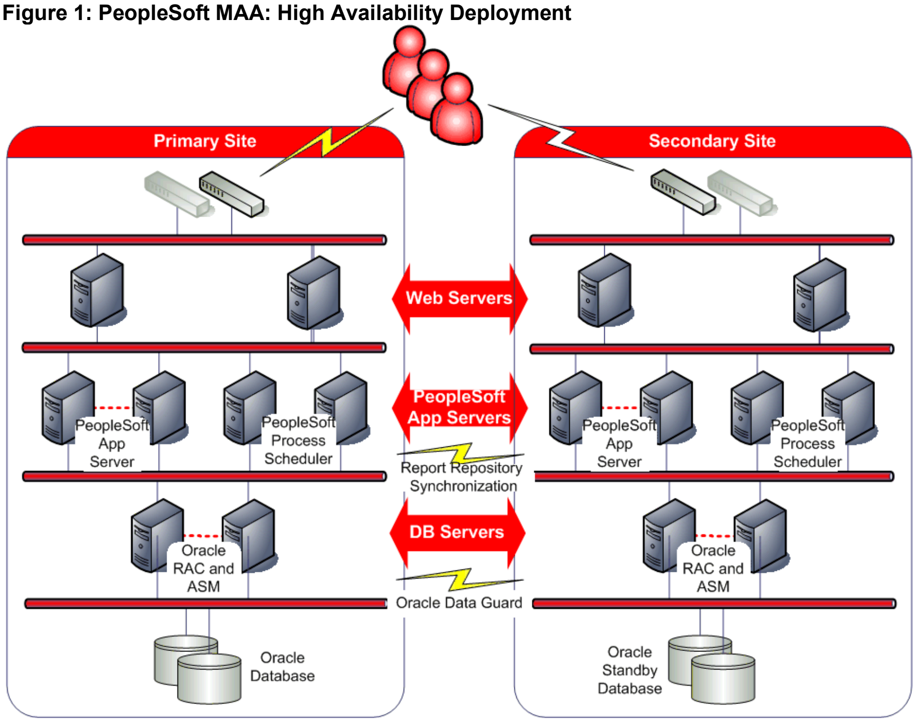
Oracle Best Practices For High Availability

Introduction

This paper describes the PeopleSoft Maximum Availability Architecture and the required operations and configuration practices to maximize PeopleSoft availability against unplanned outages and minimize downtime for planned maintenance activities. This paper also describes how recent enhancements in PeopleSoft enable faster application failover and reporting offloading to Active Data Guard.

Figure 1 shows a PeopleSoft MAA deployment with a highly redundant, resilient, and scalable primary site and an equally configured secondary site.

Figure 1: PeopleSoft MAA: High Availability Deployment



The application and database tiers are separated in the PeopleSoft application architecture allowing for configuration independence. Transitioning to MAA database requires no changes to the PeopleSoft application. In particular, it is possible to setup and configure the PeopleSoft database in an MAA configuration—including Oracle RAC, Oracle ASM and Oracle Data Guard—using the standard documentation and best practices. By implementing Oracle Data Guard, it is possible to transition to Oracle RAC and Oracle ASM with almost no PeopleSoft application downtime.

The PeopleSoft MAA configuration minimizes downtime for corruptions, database failures, site failures, and various planned maintenance activities. The secondary site can also be used to offload backups, testing, and read-only activities if the Oracle Active Data Guard option is used.

Recommendations for PeopleSoft Maximum Availability

The following list outlines the key recommendations for PeopleSoft MAA:

- Deploy the PeopleSoft database per pillar in an Oracle Database MAA configuration.
- Deploy the PeopleSoft application in a PeopleSoft MAA configuration.
- Establish a secondary standby PeopleSoft site for disaster recovery, testing, and other planned maintenance activities.
- Configure a local standby database and setup Data Guard fast-start failover:
  - Configure a local standby database to minimize downtime from data corruptions, database failures, or during planned maintenance activities such as applying patch sets, system changes, and database upgrades.

    Using a local standby database with PeopleSoft is described in the "Reducing PeopleSoft Downtime Using a Local Standby Database" [2] white paper, which is a companion to this document.

    As of Oracle 11*g* Release 2 (11.2) it is now possible to apply software patch set updates and release upgrades to the standby database first. This further reduces downtime and provides a mechanism for validating and falling back if necessary. See My Oracle Support ID 165700.1 Data Guard Standby-First Apply for further details.
  - Enable Oracle Data Guard fast-start failover with integrated application failover allow for automatic fail over and bounded recovery time of seconds or minutes.

This white paper describes each recommendation in detail and the solutions that are available for planned and unplanned outages. The MAA testing and documentation presented in this white paper is based on PeopleSoft Human Capital Management Version 9.1, PeopleTools Versions 8.50 and 8.51 running Oracle Database 11*g* Release 1 (11.1) and Oracle Database 11*g* Release 2 (11.2).

Failover Behavior

Table 1 summarizes the PeopleSoft behavior during Oracle RAC or Data Guard failover when client failover is configured. Except for a short pause as the failover occurs, the failure is transparent to the end user in most cases.

TABLE 1. PEOPLESOFT FAILOVER BEHAVIOR DURING ORACLE RAC OR ORACLE DATA GUARD FAILOVER

PEOPLESOFT CLIENT OPERATIONS	BEHAVIOR
Web client user is updating data and submits or saves the updates during or just after the database failure.	Oracle reconnects and reconstructs the database session on a surviving node and PeopleSoft resubmits the update.
Web client user is paging through queried data when the database failure occurs.	Oracle reconnects and reconstructs the database session on a surviving node. Pages are rendered from pre-fetched result-set.
Web client user is issuing a new query or switching screens just after the database failure.	Oracle reconnects and reconstructs the database session on a surviving node.

Table 2 summarizes the failover behavior of the PeopleSoft batch Process Scheduler, Application Engine (AE) jobs, Structured Query Report (SQR), PeopleSoft Query (PSQuery), XML Publisher (XMLP), and COBOL programs.

TABLE 2. PEOPLESOFT FAILOVER BEHAVIOR DURING CLIENT BATCH OPERATIONS

PEOPLESOFT CLIENT BATCH OPERATION	BEHAVIOR
Process Scheduler	Oracle reconnects and reconstructs the session on a surviving node. The process scheduler fails over with no administration intervention required.
Application Engine (AE) job submitted just *BEFORE* primary instance failure	Oracle reconnects and reconstructs the session on a surviving node but AE jobs may fail and show up in the PeopleSoft Process Monitor as “No Success”. These jobs must be resubmitted. If the AE job has been implemented to be restartable, then the process scheduler will automatically restart the job[1].
Application Engine (AE) submitted *during* or just *after* primary instance failure	Oracle reconnects and reconstructs the session on a surviving node, the AE job is then submitted on the new primary database and completes successfully.
COBOL jobs just *before* primary instance failure	If the COBOL program runs pure queries (SELECT statements), then it will fail over to the surviving node and complete successfully. If the COBOL program executes INSERTs, UPDATEs, and DELETEs, it will *not*

[1] If the AE job was not in an open transaction and the job was performing only SELECT statements, then it will fail over and complete successfully.

TABLE 2. PEOPLESOFT FAILOVER BEHAVIOR DURING CLIENT BATCH OPERATIONS

	complete successfully on the surviving node. Manual intervention is required to restart the COBOL jobs.
Crystal and SQR reports	The behavior is the same as COBOL
PSQUERY, Tree Viewer, XMLP Viewer	These PeopleSoft components will fail over and complete successfully.

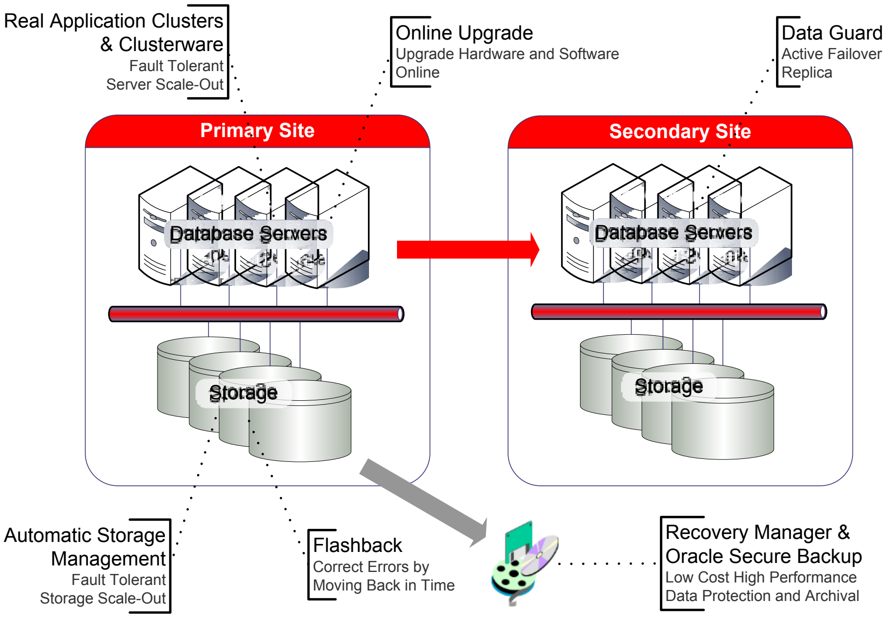
Oracle Database MAA

To maximize PeopleSoft availability, Oracle recommends deploying PeopleSoft on an Oracle Database MAA foundation that includes the following technologies:

- Oracle Real Application Clusters (Oracle RAC) and Oracle Clusterware
- Oracle Data Guard
- Oracle Flashback
- Oracle Automatic Storage Management (Oracle ASM)
- Oracle Recovery Manager (RMAN) and Oracle Secure Backup

Figure 2 shows the Oracle Database MAA configuration and technologies.

Figure 2: Oracle Database MAA: High Availability Technologies



See the *Oracle Database High Availability Overview* [3] for a thorough introduction to Oracle Database high availability products, features, and best practices.

Oracle Real Application Clusters and Oracle Clusterware

Oracle Real Application Clusters (Oracle RAC) allows the Oracle database to run any packaged or custom application unchanged across a set of clustered nodes. This capability provides the best availability for node and instance failures and most planned maintenance activities, and the most flexible scalability. If a clustered node fails, the Oracle database continues 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].

Oracle Clusterware is a cluster manager that is designed specifically for the Oracle database. In an Oracle RAC environment, Oracle Clusterware monitors all Oracle resources (such as database instances and listeners). If a failure occurs, then Oracle Clusterware automatically attempts to restart the failed resource. During outages, Oracle Clusterware relocates the processing performed by the inoperative resource to a backup resource. For example, if a node fails, then Oracle Clusterware relocates the database services being used by the application to a surviving node in the cluster. See also: *Oracle Clusterware Administration and Deployment Guide* [6].

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 Oracle Data Guard physical and logical standby databases, Oracle Active Data Guard, and snapshot standby databases.

See the following documents for complete information:

- *Oracle Data Guard Concepts and Administration* [7] for complete details about Oracle Data Guard and standby databases
- *Oracle Data Guard Broker* [8] for information about broker management and fast-start failover

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 using a 'rewind button' for your database. Oracle Flashback also quickly returns a previously primary database to standby operation after a Data Guard failover, thus eliminating the need to recopy or reinstantiate the entire database from a backup. See also Oracle Flashback Technology.

Oracle Automatic Storage Management (Oracle ASM)

Oracle Automatic Storage Management (Oracle 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, Oracle ASM spreads files across all available storage. To protect against data loss, Oracle 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 the *Oracle Database Storage Administrator's Guide*.

Oracle Recovery Manager and Oracle Secure Backup

Oracle 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 the *Oracle Database Backup and Recovery User's Guide*.

Oracle Secure Backup is a centralized tape backup management solution providing secure, high performance, and heterogeneous data protection in distributed UNIX, Linux, Windows, and Network Attached Storage (NAS) environments. By protecting file system and Oracle Database data, Oracle Secure Backup provides a complete tape backup solution for your IT environment.

Oracle Secure Backup is tightly integrated with RMAN to provide the media management layer for RMAN. See also the *Oracle Secure Backup Administrator's Guide*.

Configuring Client Failover

PeopleSoft supports seamless client failover, which allows PeopleSoft to fail over its 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 high-level steps to configure client failover are:

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.

Note: The configuration requires the following software:

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

The step-by-step instructions in the next 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 C.

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, which are enabled for transparent client failover on the primary database (PSFT).

```
srvctl add service -d PSFT -s HCM -r "PSFT1,PSFT2"
-m BASIC -e SELECT -q TRUE -l PRIMARY -z 180 -w 1
srvctl add service -d PSFT -s BATCH -r "PSFT1,PSFT2"
-m 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 (`-m`) is `BASIC`.
- The client failover type (`-e`) is `SELECT`, which allows a `SELECT` statement to fail over after 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.

Assuming your standby database is called PSFT_STBY, add services as follows:

```
srvctl add service -d PSFT_STBY -s HCM -r "PSFT1,PSFT2"
-m BASIC -e SELECT -q TRUE -l PRIMARY -z 180 -w 1
srvctl add service -d PSFT_STBY -s BATCH -r "PSFT1,PSFT2"
-m BASIC -e SELECT -q TRUE -l PRIMARY -z 180 -w 1
```

Note that PSFT and PSFT_STBY instances are running on separate host servers and clusters so the instance names (PSFT1 and PSFT2) can be used for both.

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 service 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))))

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

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 an 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 ID 876292.1. PeopleSoft release 8.50.09 is the minimum patch release that contains the FAN functionality.

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 Configure 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.

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

Step 4 Configure the PeopleSoft Application and Process Scheduler

Use the PSADMIN utility to:

- Set DBName in the Application Server Domain configuration.

Set DBNAME to the TNS alias that you configured in Step 1 above for the primary database. For example, using the names in our running example, you would set DBNAME to be "PSFT".

- Configure the Process Scheduler also using the PSADMIN utility.

  This can use the PSFT TNS alias in Step 1 or, a separate BATCH TNS alias that connects to the BATCH service as shown in Step 1 above. See the PeopleTools documentation for more information.

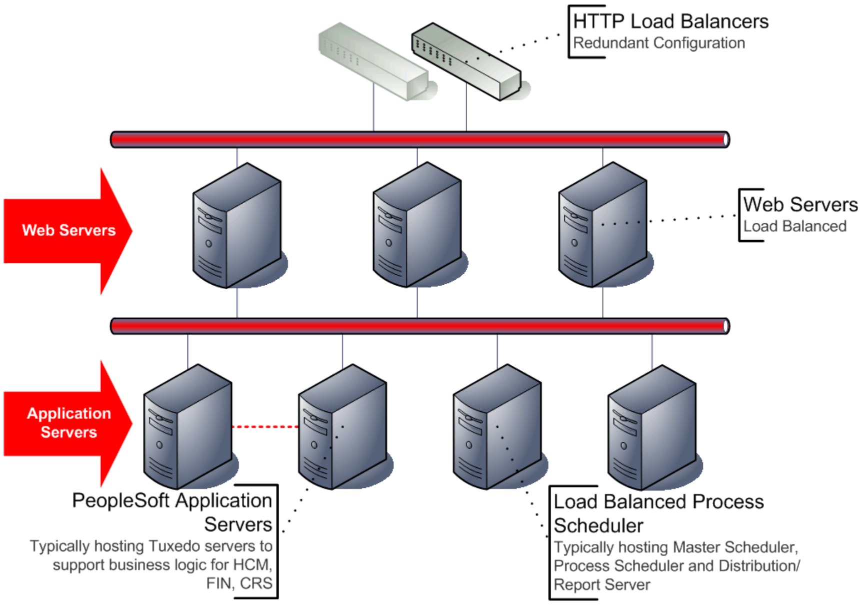
Note: PeopleTools version 8.51 and higher now supports Oracle Active Data Guard. The PSADMIN utility includes a configuration option for the secondary database TNS alias. Do not use the above TNS alias here. See the "Configuring PeopleSoft Application Server for Active Data Guard" section.

PeopleSoft High Availability Deployment

This section discusses the high availability deployment of the PeopleSoft application software that is layered on top of the Database MAA foundation.

Figure 3 shows the PeopleSoft high availability deployment in which PeopleSoft components are installed and deployed on multiple servers, and run in an active/active configuration for high availability and scalability purposes. A client-initiated workload is distributed across multiple component instances running on multiple servers through load balancing. The Web Server load is distributed by an HTTP hardware load balancer.

Figure 3: PeopleSoft MAA: Components in a High Availability Deployment


HTTP Load Balancers
Redundant Configuration
Web Servers
Web Servers
Load Balanced
Application Servers
PeopleSoft Application Servers
Typically hosting Tuxedo servers to support business logic for HCM, FIN, CRS
Load Balanced Process Scheduler
Typically hosting Master Scheduler, Process Scheduler and Distribution/ Report Server

For high availability and scalability, implement the following PeopleSoft component deployment options:

- Deploy third-party load balancers in a redundant configuration.

  Web servers and many PeopleSoft server components can be load balanced. The load balancer monitors the servers and improves availability by routing traffic appropriately when outages occur. A third-party hardware load balancer is recommended to balance the Web server load. See your load balancer documentation for details about configuring load balancing with session persistence.

- Configure two or more separate middle-tier servers and each should have two or more PeopleSoft Application Server domains.

  The application tier can be implemented with multiple physical middle-tier servers, each having one or more PeopleSoft application domains. Load balancing across all application server domains on all physical servers is achieved using a weighted load balancing and failover algorithm on top of JOLT at the Web server. See the PeopleBooks *Configuring JOLT Failover and Load Balancing* documentation for further details.

- Configure PeopleSoft with multiple batch process schedulers running on separate servers controlled by a Master Scheduler.

  To support high availability, configure PeopleSoft with multiple batch process schedulers running on separate servers controlled by a Master Scheduler. The Master Scheduler also provides load balancing of queued work across available process schedulers. When the process scheduler is running on more than one server, the Master Scheduler runs in an active-passive mode, where one Master Scheduler process runs on one of the nodes. If the node hosting the Master Scheduler fails, then the passive Master Scheduler running on a separate server assumes control.

- Install and configure the PeopleSoft Pure Internet Architecture (PIA) in a directory outside the application server `PS_HOME`.

  The PeopleSoft Distribution Server Report Repository requires a file system, which is used to store file attachments, published reports, and other documents in the PeopleSoft application. The file system is accessed in parallel by all Process Scheduler Servers and is a critical part of the PIA and so must be deployed in a highly available configuration to avoid a single point of failure. Typically, this would be achieved through a cluster file system or network attached storage (NAS) using NFS.

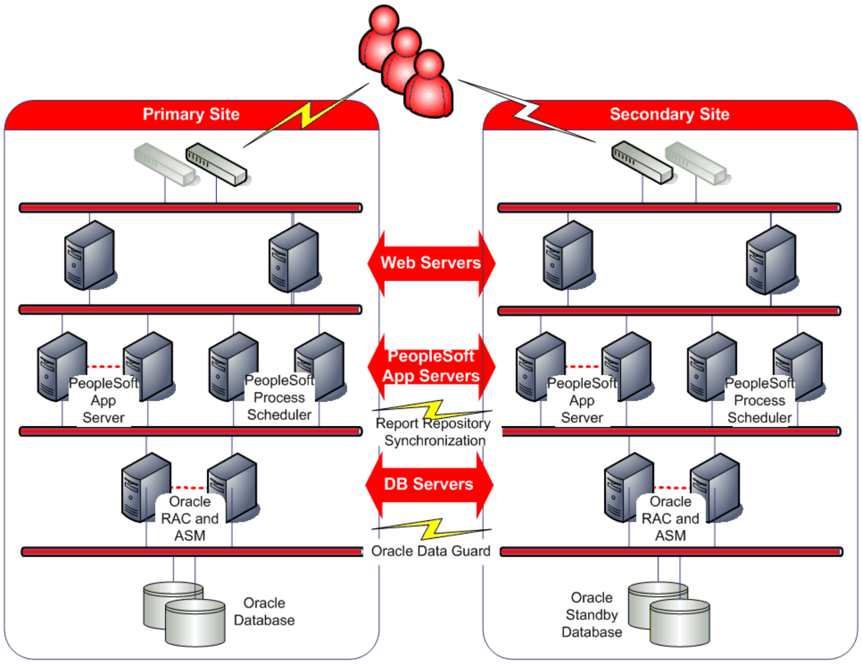
  See the “PeopleSoft Secondary Site Deployment” section in this white paper to configure the Report Repository in a directory outside of the `PS_HOME`, `PS_CFG_HOME`, and the PIA directories on any shared or NFS file system.

PeopleSoft Secondary Site Deployment

This section describes how to establish a **secondary site** to guard against an entire site failure and to reduce downtime during certain planned maintenance operations.

Figure 4 shows an example of primary and secondary sites for a PeopleSoft MAA deployment.

Figure 4: PeopleSoft MAA: Secondary Site Deployment



For high availability and scalability, implement the following recommendations when deploying a secondary site:

- To achieve the same service levels after a failover, the size and configuration of the secondary site should replicate that of the primary site.

  Note: Although it is technically possible to deploy a reduced configuration on the secondary site, this is not recommended.

- After the secondary site is established, perform routine role transitions (Data Guard switchovers or failovers) to make sure that the secondary site is operational and viable failover target in the case of a real emergency.

The following sections describe:

- How to establish the Oracle Data Guard standby database and a PeopleSoft Enterprise environment on the secondary site.
- How to perform switchover, failover, testing, and automated startup.

Deploying the Oracle Data Guard Standby Database

Perform the following steps to establish a Data Guard standby database:

Step 1 Install and configure Oracle Clusterware, Oracle ASM, and Oracle Database on the database tier at the standby site.

Step 2 Backup, transport, and restore the database files to the standby site.

You may also use the RMAN `DUPLICATE` command to instantiate a standby database, if appropriate. See the "Oracle Data Guard 11*g* Installation and Configuration Best Practices on Oracle RAC" MAA white paper

Step 3 If the standby database at the secondary site will be configured as an Oracle RAC database, then configure the standby with the same database services used on the primary database.

Use the `SRVCTL` commands, as shown in the "Configuring Client Failover" section earlier in this white paper. Doing so updates the Oracle Cluster Repository (OCR) at the secondary site.

Step 4 Assess the wide area network (WAN) bandwidth to properly configure the Oracle Data Guard redo transport services.

See the "Properly Configure TCP Send / Receive Buffer Sizes" section in the *Oracle Database High Availability Best Practices 11g Release 1 (11.1)* [4] documentation.

Step 5 Use the Data Guard broker to simplify configuring Data Guard, and enable the broker configuration to activate Data Guard.

See the *Oracle Data Guard Broker* [8] documentation for the detailed steps.

Establishing the Secondary PeopleSoft Enterprise Deployment

To complete the standby site, install PeopleSoft and establish a secondary PeopleSoft enterprise for each pillar configured at the primary site. The secondary PeopleSoft enterprise is configured to connect to the standby database. In the event of a switchover or failover to the standby site, the standby database is opened and the secondary PeopleSoft enterprise is started.

Perform the following high-level steps to set up the PeopleSoft Enterprise at the secondary site:

Step 1 Install the WebLogic Web Server software on each of the Web tier servers.

Step 2 Install the Oracle Database client software to be used by the PeopleSoft Application Server on each application-tier server.

Step 3 Configure the `TNSNAMES.ORA` file on each of the application server tier nodes using the same TNS alias names used in the primary environment, but list the nodes for the standby database.

If the standby database is configured for Oracle RAC, include all standby Oracle RAC nodes in the ADDRESS descriptor. The SERVICE_NAME for each alias should be the same as what was used on the primary database.

Step 4 Create the PS_HOME and PS_CFG_HOME directories for the application tier servers similar to that of the primary site.

Step 5 On the Web Servers, create the directory for the Report Repository where the Distribution Server will place published reports files.

This directory should have the same mount point name and path as the mid tiers at the primary site. The Report Repository directory should be separate from the PS_HOME, PS_CFG_HOME, and PIA directories when using a shared file system. See the "Synchronizing the Report Repository" section in this white paper.

Step 6 Install the same version of PeopleTools as on the primary site for each of the application-tier server nodes.

Step 7 Install the same version of the PeopleSoft application (for example, HCM, Financials, and so on) on each application server node.

Step 8 Perform PeopleTools configuration (using the PSADMIN utility) on each of the application tier nodes.

Important: Use the TNS alias name from step 3 above to set the DBName in the PeopleTools PSADMIN configuration utility. Again, the TNS alias names must be the same as that on the primary database.

Step 9 Install the COBOL compiler and runtime environment for the servers that will host the PeopleSoft Process Scheduler and Application Engine.

Step 10 Synchronize the Report Repository directory from the primary site. For example, use the RSYNC command. See the "Synchronize the Report Repository" section later in this white paper.

Step 11 Do not replicate (such as with the RSYNC command) the PS_HOME, PS_CFG_HOME, or the PIA directories from the primary site because that would overwrite site-specific configuration files.

Step 12 During a planned maintenance period, perform a Data Guard switchover of the database from the primary site to the standby site. Once completed, the standby database should be running in the primary role.

Step 13 Start up the PeopleSoft Application Server and Web Server and perform any further site-specific configuration and testing, as necessary.

Step 14 Once you have completed testing in Step 13, then switch back to the original site by shutting down the PeopleSoft Application and Web Servers.

Step 15 Perform a Data Guard switchover to return the databases back to their original primary and standby roles, and then start the primary PeopleSoft Application and Web Servers.

Synchronize the Report Repository

The PeopleSoft Distribution Server Report Repository (file system) contains critical report files that must be made available on the standby site in the event of a disaster. To ensure that the Report Repository is available after failover, maintain a standby copy of the file system containing the report repository using a replication method (such as `rsync`).

The Report Repository is defined when installing and configuring the PeopleSoft PIA Web server component. The best practice is to install the Web servers on separate servers from the PeopleSoft Application Servers. On the Web servers, the `CONFIGURATION.PROPERTIES` file that contains the `ReportRepositoryPath` setting can be found where PIA has been installed. For example:

```
/u01/app/webserv/peoplesoft/applications/peoplesoft/PORTAL.war/WEB-INF/psftdocs/ps
```

Although there are several file copying tools available to create and maintain the copy, consider using the `RSYNC` command, which is a readily available utility that is well suited to maintaining copies over a wide area network. It will be necessary to switch the direction of synchronization in the event of a switchover or failover.

DNS Push after Site Failover

After a site failover, PeopleSoft end users must connect to the standby site to access the PeopleSoft application. To make the switchover seem transparent to the end users, implement a Domain Name Server (DNS) push. In a DNS push, the IP address associated with the PeopleSoft service is changed in DNS and then propagated to the end user's browser. Then, when the user tries to connect, they pick up the new address and are routed to the alternate location.

Configuring PeopleSoft Application Server for Active Data Guard

PeopleSoft PeopleTools version 8.51 supports Oracle Active Data Guard from within the PeopleSoft Application Server. This capability allows report requests to be routed to a physical standby database that is also configured with the real-time query feature of Oracle Active Data Guard. Reports execute on the Active Data Guard database instead of the primary OLTP database, obtaining near real-time results. The lag time for the Active Data Guard Database is configurable, but 10 to 60 seconds is a typical lag time range. The Active Data Guard Database can be configured for either a local or remote physical standby database.

The Active Data Guard physical standby database can still serve as a failover target. Thus, a dual purpose is now realized, giving higher utilization of the standby database while still maintaining a disaster recovery site.

This section describes the step-by-step procedures for configuring the PeopleSoft Application Server for use with Active Data Guard. These procedures are specific to Oracle Database 11*g* Release 2 (11.2). See Appendix C for instructions on configuration with Release 11.1.

Perform the following steps:

Step 1 Configure and enable Active Data Guard

This step assumes that you have a physical standby database and Data Guard broker is already configured. Your primary database is shipping redo to the standby database (not archivelog shipping).

a) Log into Data Guard Broker and verify the database is shipping redo by issuing the following command:

```
DGMGRL> show database verbose <physical standby database
name>;
```

```
Intended State:  APPLY-ON
Real Time Query: ON
```

If the output displays `Real Time Query: ON`, as shown in the example, then proceed to step b). Otherwise, open the database read-only and enable real-time query, as follows.

- Log onto SQL*Plus and issue the following command:

```
SQL> alter database open read only;
```

```
DGMGRL> show database verbose <physical standby
database name>;
```

- Reissue the `SHOW DATABASE VERBOSE` command and verify that real-time query is enabled and the intended state is `APPLY-ON`.

b) Issue the following query to monitor the apply lag of the Active Data Guard database:

```
SELECT name, value, datum_time, time_computed
FROM v$dataguard_stats
WHERE name = 'apply lag';
```

The `value` column is the current lag time in seconds behind the primary database. For further details, see the *Oracle Data Guard Concepts and Administration* [7].

Step 2 Create a new database service for accessing Active Data Guard.

Use the SRVCTL utility to add the service:

a) Add the service to the Oracle RAC primary database. For example, the following example adds the PSQUERY service:

```
srvctl add service -d PSFT -s PSQUERY -r "PSFT1,PSFT2" -m
BASIC -e SELECT -q TRUE -l "PHYSICAL_STANDBY" -z 180 -w 5
```

b) Add the same service to the standby database:

```
srvctl add service -d PSFT_STBY -s PSQUERY -r
"PSFT1,PSFT2" -m BASIC -e SELECT -q TRUE -l
"PRIMARY,PHYSICAL_STANDBY" -z 180 -w 5
```

The SRVCTL commands assign two database roles to the `PSQUERY` service on the `PSFT_STBY` database: `PHYSICAL_STANDBY` and `PRIMARY`. This ensures that this service will always be available on this database even after a role transition.

The service (`PSQUERY`) still needs to be created on the primary database so the service definition can be propagated through the redo to the standby. You cannot create this service on the physical standby because it is read-only.

c) On the primary, invoke SQL*Plus, log in as SYSDBA, and execute the following statement:

```
EXECUTE DBMS_SERVICE.CREATE_SERVICE('PSQUERY', 'PSQUERY',
NULL, NULL,TRUE, 'BASIC', 'SELECT', 180, 5, NULL);
```

The parameters passed in the `DBMS_SERVICE.CREATE_SERVICE` procedure must match those specified in the above SRVCTL command:

d) Start and stop the service on the primary:

```
srvctl start service -d PSFT -s PSQUERY
srvctl stop service -d PSFT -s PSQUERY
```

This is required for SRVCTL to perform the check and service startup as it uses the `DBMS_SERVICE` package.

e) Start the service on the standby database:

```
srvctl start service -d PSFT_psftstby -s PSQUERY
```

The service should start on the standby. You can check this by using the LSNRCTL utility to verify that the service is registered and listed with the listener:

```
lsnrctl status
```

If the SRVCTL utility reports errors in Step 2a above, then this is most likely due to a mismatch of values between the SRVCTL command and the `DBMS_SERVICE.CREATE_SERVICE()` procedure call.

Step 3 Create TNS connect string

The PeopleSoft application server requires a separate connect string that connects to the Active Data Guard database for reporting. In the following example, the connect string `PSQUERYS` connects to the `PSQUERY` service:

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

Ensure that the TNS alias is accessible by the PeopleSoft application server. Test access to the Active Data Guard database using SQL*Plus. You will use this TNS alias in the next step for the “secondary” database. At this point, you are ready to configure the PeopleSoft application server for Active Data Guard.

Step 4 Configure the PeopleSoft Application Server

a) Complete the procedures described in the *PeopleTools 8.51 PeopleBook: Data Management* [10] documentation. Perform all procedures on the primary database. Do not shut down the Active Data Guard database while following these procedures.

b) When the PeopleSoft documentation instructs you to create the database link, ensure the database link meets these requirements:

- Contains TNS connect alias that points only to the primary database service.
- Uses `TNSNAMES.ORA` file with the aforementioned TNS connect alias on all databases involved in the Data Guard configuration.

- Connects to the `SCHEMA` on the primary database as directed by the PeopleSoft documentation.

If any requirements are not met, users who submit reports to the Active Data Guard database via the PeopleSoft Web user interface may receive errors. After switchover or failover where the Active Data Guard database is not available, the database link will point back (loopback) to the same database to prevent any outage of any reports submitted to the `PSQUERY` service that would now be running on the primary database. See the "Best Practices for PeopleSoft with Active Data Guard" section for further details.

Operational Best Practices for PeopleSoft to Use Active Data Guard

Implement the following best practices when configuring PeopleSoft to use an Active Data Guard database:

- Maintenance on the Active Data Guard Database: If maintenance is to be performed on the Active Data Guard database and it requires the database or service to be unavailable, then start the service on the primary database before shutting down the service and/or Active Data Guard database.
- Snapshot standby: If the Active Data Guard database is to be converted to a snapshot standby, then start the PeopleSoft services on the primary database before converting the Active Data Guard to a snapshot standby database.. When converting the database back to a physical standby for Active Data Guard, it will be necessary to restart the read-only service (PSQUERY) before PeopleSoft can make use of it. Once started, this service can be shutdown on the primary.
- Fast-start failover: If the Active Data Guard database is the failover target for Data Guard fast start failover, then ensure that the service used by PeopleSoft is configured to be available for both database roles: `PHYSICAL_STANDBY` and `PRIMARY`, as described in Step 2 of the "Configuring PeopleSoft Application Server for Active Data Guard" section.
- Multiple physical standby databases: If your environment contains multiple physical standby databases configured as an Active Data Guard reader farm, ensure the TNS alias lists all the hosts (or Single Client Access Names) that will be supporting the read-only service. In such an environment, you need not start the service on the primary post failover or switchover, but the database link on all reader farm databases *must* always point back to the primary.

Performing Switchover, Failover, and Testing

Use the following step-by-step procedures to perform switchovers, failovers, and to test the validity of the standby database.

Switchover Procedure

Perform the following steps:

1. Verify the standby database is up-to-date and operating correctly.
2. Shut down PeopleSoft on the primary site.
3. Switch over to standby database.
4. Start the original standby in the primary database role. By using the Data Guard broker in step 3 to perform the switchover, you will have completed this step automatically.
5. Ensure the standby PeopleSoft Distribution Server Report Repository file system is up-to-date and reverse the synchronization direction. See the documentation specific to the replication tool you are using.
6. Start the PeopleSoft application and Web servers at the standby site.

See the *Oracle Data Guard Concepts and Administration* [7] documentation for detailed database switchover steps and commands. Follow the same steps to switch the databases back to their original roles.

Failover Procedure

Perform the following steps:

1. Ensure that Flashback Database is enabled at the primary and standby databases.
2. Fail over to standby database.
3. Open the standby database.
4. Start PeopleSoft.
5. When the primary site becomes available, reinstate the old primary database by following the Data Guard scenario for reinstating an old primary. See the documentation specific to the replication tool you are using.

See the *Oracle Data Guard Concepts and Administration* documentation [7] for detailed database failover steps.

Note: These procedures assume that all of the Oracle Database MAA best practices have been implemented, including the use of Oracle Flashback Database. Enabling Oracle Flashback Database allows you to perform step 5 without a full database restore.

Testing and Restoring the Standby Database

You can verify the viability of the standby site anytime while the primary site remains in operation. When you complete your validation, then you can quickly flashback the standby and resume recovery, using Oracle Flashback Database.

This procedure assumes the primary site remains active and is running in the primary role, and the standby site is in physical standby mode and applying redo.

1. Cancel Redo Apply on the standby database.
2. Convert the standby database to a snapshot standby.
3. Perform testing.
4. Convert the snapshot standby back to a physical standby and resume standby operation.

See the *Oracle Data Guard Concepts and Administration* [7] documentation for detailed failover steps.

Note: With Oracle Active Data Guard, you can query the standby database while it's recovering.

PeopleSoft Application Patching and Maintenance Procedures

Once in place, a secondary site must be kept up-to-date with the primary. Use the following best practices to maintain availability during patching and maintenance procedures:

- When you change the PeopleSoft software and configuration on the primary site, you must apply the same changes to the secondary site.
  - **Database updates:** Database changes are propagated automatically by Oracle Data Guard. PeopleSoft Report Repository changes are propagated to Standby PeopleSoft Report Repository by using the appropriate replication method (such as the `rsynch` mechanism described earlier).
  - **Software patching:** If the PeopleSoft Application Server or Web Server software is patched on the primary site, then you must perform the same patching procedures on the secondary site. Because the `PS_HOME` and `PS_CFG_HOME` directories are not being synchronized, the site-specific configurations are preserved, yet the primary and secondary sites are kept up-to-date.
- Upgrade the software on the secondary site without updating data on the Oracle database.
- Avoid or minimize the time when the secondary site is not synchronized or the different parts of the secondary site are not synchronized with each other.

  For example, in a PeopleSoft application upgrade scenario, it may be necessary to suspend Oracle Data Guard managed recovery and Standby PeopleSoft Report Repository synchronization until an upgraded Standby PeopleSoft Enterprise has been established.

Automating Management Tasks with Oracle Data Guard Broker

The "Oracle Data Guard" section earlier in this white paper described management using the Oracle Data Guard broker. All management operations can be performed locally or remotely through either of the broker's easy-to-use interfaces:

- The Data Guard management pages in Oracle Enterprise Manager, which is the broker's graphical user interface (GUI).
- The Data Guard command-line interface called DGMGRL.

Using the broker, you can automate the entire switchover and failover process, including automatically triggering a failover. Typically, automated switchover or failover proceeds as described in the following list, but you can omit certain tasks (such as automatic triggering) if a task does not make sense for your implementation:

1. Either fast-start failover determines that a failover is necessary and initiates a failover to the standby database automatically, or the database administrator issues a broker command to initiate a switchover or failover.
2. When the database switchover or failover has completed, the `DB_ROLE_CHANGE` database event fires.
3. The event causes a trigger to be fired, which calls a script that configures and starts the PeopleSoft application.

Perform the following steps to implement fast-start failover:

> **Note:** The procedure assumes that the primary site is in live operation and the standby site is in standby mode and applying redo data:

Step 1 Develop a PeopleSoft startup script.

1. Develop a script that automates the PeopleSoft startup process. See Appendix B for script examples that you can modify to suit your environment and requirements.
2. Make sure `ssh` (or equivalent) is configured so that remote shell scripts can be executed without password prompts.
3. Make sure that the operating system user has the appropriate permissions to execute the script.

Step 2 Automate script execution using a trigger.

Create a database event `DB_ROLE_CHANGE` trigger that fires after a failover changes the role of the database role from standby to primary database. For example:

```
CREATE OR REPLACE TRIGGER ps_fsfo
AFTER DB_ROLE_CHANGE ON DATABASE
DECLARE
  v_db_unique_name varchar2(30);
BEGIN
  select upper(VALUE) into v_db_unique_name
  from v$parameter where NAME='db_unique_name';

-- Place a check here to determine if this
-- is a primary site vs.
-- a remote DR site.
--
-- If you have a local standby
-- in addition to a remote DR
-- site standby, and you have client failover/FAN
-- set up for a local standby database,
-- you don't want this trigger
-- to attempt to start any of the
-- PeopleSoft middle tier components.
--
-- In this example:
-- PSFT is at primary site
-- PSFT_STBY is at secondary site

  IF v_db_unique_name = 'PSFT_STBY'
     OR v_db_unique_name = 'PSFT' THEN

       dbms_scheduler.create_job(
            job_name=>'postover',
            job_type=>'executable',
            job_action=>
      '/home/oracle/FSFO/' || v_db_unique_name||'_fsfo.sh',
           enabled=>TRUE
       );
  END IF;
END;
/
```

The trigger calls a wrapper script named <*db_unique_name*>_fsfo.sh which in turn calls the fsfo.sh script. This script is necessary because it is not possible to directly pass arguments to a script from DBMS_SCHEDULER. You must create a wrapper script for the primary and standby databases.

The following example creates a script named PSFT_fsfo.sh for the primary site:

```
#!/bin/sh
/home/oracle/FSFO/fsfo.sh PSFT
```

The following example creates a script named PSFT_STBY_fsfo.sh for the standby site:

```
#!/bin/sh
/home/oracle/FSFO/fsfo.sh PSFT_DR
```

Step 3 Configure fast-start failover.

See the Oracle Data Guard Broker for 11*g* Release 2 documentation for details about how to configure fast-start failover.

Planned and Unplanned Outage Solutions

The following sections lists the outages that may occur in a PeopleSoft environment and the Oracle solution you would use to keep application downtime to a minimum. In all cases, we are focused on PeopleSoft Application downtime as perceived by the end user, not the database downtime.

Unplanned Outage Solutions

Table 3 describes the unplanned outages that may be caused by system or human failures in a PeopleSoft environment and the technology solutions you would use to recover and keep downtime to a minimum.

TABLE 3. UNPLANNED OUTAGES AND SOLUTIONS

OUTAGE TYPE	ORACLESOLUTION	BENEFITS	RECOVERY TIME
PeopleSoft node or component failure	Hardware load balancing	Surviving nodes pick up the load	The time it takes the affected users to reconnect
	Distributed services across Tuxedo Application Servers	Surviving nodes continue processing	No downtime
Database node or instance failure	Oracle RAC and FAN/client failover	Automatic recovery of failed nodes and instances, transparent application failover	Seconds (transparent to users) [2] Updates may need to be resubmitted
Site failure	Oracle Data Guard	Fast-start failover[3]	Seconds to minutes (requires DNS push and application startup)
Storage failure	Oracle ASM	Mirroring and automatic rebalance	No downtime
	Oracle Data Guard	Fast-start failover	Seconds to minutes[2]

[2] You may need to manually restart some AE and COBOL programs.

[3] For the secondary site you should use the `ASYNC` transport. If you only want to use the `SYNC` transport, then fast-start failover is not usually viable if the latency is high because of possible response time and throughput impact.

TABLE 3. UNPLANNED OUTAGES AND SOLUTIONS

	RMAN with flash recovery area	Fully managed database recovery and disk-based backups	Hours to days (depending on the extent of the storage failure)
Human error	Oracle Flashback	Database and fine-grained rewind capability	Minutes, but the actual time is dependent on the detection time and executing repair solution with Flashback Technologies
	LogMiner	Log analysis	Minutes to hours
Data Corruption	Oracle Data Guard	Automatic validation of redo blocks before they are applied, fast-start failover to an uncorrupted standby database. With Oracle Active Data Guard release 11.2, block corruptions on the primary can be repair automatically using good blocks on the standby without any application downtime.	Seconds to minutes
	RMAN with flash recovery area	Online block media recovery and managed disk-based backups	Minutes to hours

Planned Outage Solutions

Table 4 describes the planned outages in a PeopleSoft environment and the technology solutions you would use to keep downtime to a minimum.

TABLE 4. PLANNED OUTAGES AND SOLUTIONS

MAINTENANCE ACTIVITY	SOLUTION	PEOPLESOFT OUTAGE
Mid-Tier operating system or hardware upgrade	Hardware Load balancing, distributed services across Tuxedo Application Servers	No downtime
PeopleSoft application patching (application tier only)	PeopleSoft patching	Minutes to hours

TABLE 4. PLANNED OUTAGES AND SOLUTIONS

PeopleSoft application configuration change	PeopleSoft application rolling restart	Minutes
PeopleSoft upgrades	PeopleSoft upgrades	Hours to days (depending on database size)[4]
Database tier operating system or hardware upgrade	Oracle RAC	No downtime
Oracle Database interim patching	Oracle RAC rolling apply	No downtime
Oracle Database 11*g* online patching	Online patching	No downtime
Oracle Clusterware upgrade and patches	Rolling apply / upgrade	No downtime
Database storage migration	Oracle ASM	No downtime
Oracle ASM upgrade	10*g*: Oracle Data Guard	Seconds to minutes
	11*g*: Oracle ASM rolling upgrade	No downtime
Migrating to Oracle ASM or migrating a single-instance database to Oracle RAC	Oracle Data Guard	Seconds to minutes
Patch set and database upgrades	Release 11.1 or greater: Oracle Data Guard transient logical rolling upgrade	Seconds to minutes

About PeopleSoft Multiple-Pillar Deployments

PeopleSoft Enterprise Applications uses the concept of *pillars* for specific lines of business functionality. Each pillar has its own dedicated database with a dedicated set of application

[4] 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 can help you plan and execute the upgrade.

server domains and Web servers. For example, if Human Capital Management, Financials, and Enterprise Portal Management have been implemented, then there are three pillars in this deployment. Information exchanged between pillars is achieved using the PeopleSoft Integration Broker.

The PeopleSoft Integration Broker does not create distributed database transactions using Oracle Database two-phase commit. Rather, information exchanges between pillars is achieved using a message queue and is processed by the Integration Broker messaging component. The Integration Broker is configured to run on one of the pillars, and connects directly to all other pillar databases to perform its work.

For PeopleSoft Enterprise Application implementations that deploy multiple PeopleSoft pillars (HCM, FIN, SupplyChain, CRM, EPM), all of the configuration and operational best practices described in this white paper should be applied to each pillar. Each pillar can be configured with Oracle Clusterware and Oracle RAC, and should have its own Data Guard standby database at a secondary disaster-recovery site for data protection, scalability, and high availability.

Behavior for Switchover

When performing a Data Guard switchover involving pillars to a secondary disaster-recovery site, you can adopt either of the following options:

- One pillar: Shut down Integration Broker prior to perform a switchover of the pillar. Performing switchover of pillars individually provides the highest flexibility. Do not restart the Integration Broker until all pillars are running on the same site.

  Note: If only one pillar is switched over to the secondary site and you restart the Integration Broker, then the Integration Broker will fail because it cannot connect to the one pillar that has now been switched over to the secondary site. However, switching over just one pillar is a useful option for performing rolling maintenance tasks, pillar by pillar.

- Multiple pillars: Switch over all pillars at the same time to the secondary disaster-recovery site.

If you do not follow these guidelines, the PeopleSoft Integration Broker could fail to connect to its local database service.

Behavior for Failover

For a Data Guard failover, Table 5 shows how data loss is dependent on the amount of redo data received by the standby database at the time of the failover.

TABLE 5. DATA LOSS FOR PILLAR DEPLOYMENTS

PROTECTION MODE	REDO TRANSPORT	EXPECTED DATA LOSS
Maximum Protection or Maximum Availability	SYNC AFFIRM	There will be zero data loss in the event of a failover. All pillar databases will be consistent.

TABLE 5. DATA LOSS FOR PILLAR DEPLOYMENTS

Maximum Performance	ASYNC or ARCH	A failover will most likely result in data loss. The extent of the data loss is dependent on the redo transport service configuration and on the amount of redo that was not received by the standby database at the time of the failover.

The pillars involved in a failover may be inconsistent at the standby site and may require some manual intervention to resolve the inconsistencies. If the inconsistency between pillars is determined to be significant, one option is to recover each pillar database to the same point in time.

For further information on point-in-time recovery in distributed or multiple-database environment, see My Oracle Support ID 1096993.1.

Conclusion

Using the proper Oracle technologies coupled with MAA best practices, you can achieve high availability, scalability, and data protection in a PeopleSoft enterprise deployment. The Oracle Database high availability and scalability technologies: Oracle ASM, Oracle RAC, Oracle Data Guard (with the Data Guard broker), and Fast Application Notification (FAN) provide the foundations for implementing such systems. Enabling PeopleSoft for server-side client failover and enabling the acceptance of FAN events ensures that the user community has minimal impact no matter the scale of the disaster.

Furthermore, Data Guard fast-start failover plus the use of a database trigger `DB_ROLE_CHANGE` system event provides monitoring and application startup automation to reduce Recovery Time Objective (RTO). Oracle Database 11*g* Release 2 (11.2) provides additional features that simplify the configuration steps that enable high availability features, making these components more integrated and easy to implement.

Oracle continues to strive for more integration and seamless configuration in its product sets, with little to no impact to the PeopleSoft application. The MAA best practices in this white paper provide a simplified set of steps that yield high value for high availability, scalability and data protection for minimal cost.

Appendix A: Configuration Steps for PeopleSoft Application Server

Once you have configured the database services and TNSNAMES.ORA aliases, you must configure the PeopleSoft Application Server to connect to the database using these aliases and to meet PeopleSoft security requirements via the PS.PSDBOWNER table.

Perform the following configuration steps:

Step 1 Update the PS.PSDBOWNER table.

Update the PS.PSDBOWNER table to reflect the proper access to PeopleSoft applications. The TNSNAMES.ORA alias (HCM and BATCH) abstract the database access that PeopleSoft will use. As such, the security component requires the PeopleSoft PS.PSDBOWNER table reflect the abstracted database access. This table contains two columns: DBNAME and OWNER_ID. Using the above TNSNAMES.ORA example, add these two entries in the table, as follows:

1. Using SQL*Plus, log into the PeopleSoft database as SYSADM on the database server. For example:

```
SQLPLUS SYSADM/<password>
```

2. Referencing the TNSNAMES.ORA example above, issue the following INSERT statements into the PS.PSDBOWNER table:

```
INSERT INTO PS.PSDBOWNER VALUES (‘BATCH’,’SYSADM’);
INSERT INTO PS.PSDBOWNER VALUES (‘PSFT,’SYSADM’);
COMMIT;
```

Note: The TNS connection alias (not the service name) is the value used for the DBNAME column.

For further information about configuring the PS.PSDBOWNER table, see the PeopleTools Installation Guide that is specific to your PeopleTools version.

Step 2 Configure PeopleSoft to use database services.

Configure the PeopleTools layer to use the above TNSNAMES.ORA connect alias to connect to the services described in the “Configuring Client Failover” step earlier in this white paper. Use the PSADMIN tool to configure the application domain and the Process Scheduler to make use of the PSFT and BATCH connect alias respectively, as follows:

1. Log onto the nodes that are running the PeopleSoft Application Server.
2. Run the PSADMIN tool that is located under the $PS_HOME/appserv directory,.
3. Select **Application Server**.

4. Select **Administer a domain**.
5. From the list of domains, select the domain you wish to configure.
6. Select **Configure this domain**.

   This option will require the domain to be shut down. Respond with '**y**' when prompted to continue.
7. Set the value for "DBNAME" to the TNSNAMES.ORA connection alias you have defined step 6. For the example used this white paper, it would be set to "PSFT".
8. Exit from the application domain configuration and restart the application domain.
9. Follow the instructions in steps 1 through 8 except in step 3 choose **Process Scheduler.**

   Use the PSADMIN tool to configure the Process Scheduler and ensure "DBNAME" is set to the correct TNSNAMES.ORA connection alias. For the example in this white paper, it would be set to "BATCH".
10. Restart the PeopleSoft Application Server and Process Scheduler.

Step 3 Validate PeopleSoft use of Database Services.

Query the V$SESSION view, to verify that PeopleSoft is using the services. For example:

```
set pagesize 100
column process format a8
column username format a10
column service_name format a10
select process,username,program,service_name
from v$session
where machine = 'halinux13'
/
```

Substitute "halinux13" with the actual names of the nodes running your PeopleSoft Application Servers. The query will display output similar to the following example:

```
PROCESS   USERNAME     PROGRAM                                      SERVICE_NA
-------------- ----------------  -------------------------------------------------   ----------
6823       SYSADM       PSPRCSRV@halinux13 (TNS V1-V3)              BATCH
7719       SYSADM       PSSAMSRV@halinux13 (TNS V1-V3)              HCM
7747       SYSADM       PSMONITORSRV@halinux13 (TNS V1-V3)          HCM
7664       SYSADM       PSQCKSRV@halinux13 (TNS V1-V3)              HCM
6748       SYSADM       PSAESRV@halinux13 (TNS V1-V3)               BATCH
7639       SYSADM       PSAPPSRV@halinux13 (TNS V1-V3)              HCM
6729       SYSADM       PSAESRV@halinux13 (TNS V1-V3)               BATCH
6797       SYSADM       PSDSTSRV@halinux13 (TNS V1-V3)              BATCH
7610       SYSADM       PSAPPSRV@halinux13 (TNS V1-V3)              HCM
```

If you are running Oracle RAC, issue the query on each Oracle RAC instance.

Appendix B: Sample Configuration and Startup Scripts

This appendix provides the following scripts:

- fsfo.sh
- startPS.sh
- startWS.sh

The fsfo.sh Script

The following script is named `fsfo.sh` in our example and is executed by the `ps_fsfo` trigger when the trigger on the `DB_ROLE_CHANGE` system event occurs and the standby transitions to the primary database role. Only one node can perform the database role change. The `DB_ROLE_CHANGE` system event is designed to run on any database node however, only one node can perform the `DB_ROLE_CHANGE`. This script should be placed on all Oracle RAC nodes.

```
#!/bin/sh

# Enable/Disable the script,
# set value to 1 to perform the steps in the script
############################################################
ENABLED=1

# Arg1 DB_UNIQUE_NAME determines the site
# that needs to be activated.
############################################################
DB_UNIQUE_NAME=$1

# Constants, modify according to your environments
############################################################
DB_NAME=PSFT
```

```
SITE1=PSFT
SITE2=PSFT_DR

DB_NODES_SITE1="halinux11 halinux12"
DB_NODES_SITE2="rmdclinux10 rmdclinux11"

# PS = application server nodes
# WS = web server nodes
PS_NODES_SITE1="halinux13 halinux15"
WS_NODES_SITE1="halinux16 halinux17"
PS_NODES_SITE2="rmdclinux13 rmdclinux14"
WS_NODES_SITE2="rmdclinux15 rmdclinux16"

# Set OH to the Oracle Home client used by PeopleSoft Application Server.
OH=/u01/app/oracle/product/11.1.0/db_1

# The following two users can be different.  Set accordingly.
DBOSUSER=oracle
APPSOSUSER=oracle

# Logfile
###########################################################
LOGF=/home/oracle/FSFO/PeopleSoft_fsfo.log
DETAILLOGF=/home/oracle/FSFO/PeopleSoftLdetailfsfo.log
exec >>$LOGF 2>>$DETAILLOGF

# Start executing
###########################################################

echo ""
echo "----------------------------------------------------"
echo "script started at `date`"
echo "----------------------------------------------------"
echo ""

# Initialize the variables for the correct Site
###########################################################
if [ ${DB_UNIQUE_NAME}x = ${SITE1}x ]; then
  DB_NODES=$DB_NODES_SITE1
  PS_NODES=$PS_NODES_SITE1
  WS_NODES=$WS_NODES_SITE1
elif [ ${DB_UNIQUE_NAME}x = ${SITE2}x ]; then
  DB_NODES=$DB_NODES_SITE2
  PS_NODES=$PS_NODES_SITE2
  WS_NODES=$WS_NODES_SITE2
else
  echo "`date` -- Error !"
  echo "(Err) Missing/Invalid argument DB_UNIQUE_NAME:
\"$DB_UNIQUE_NAME\""
  exit 1
fi

echo "Site: $DB_UNIQUE_NAME on `hostname` as `id`"
echo "----------------------------------------------------"
echo "`date` -- Start PeopleSoft Application Server on All Nodes"
echo "----------------------------------------------------"

for node in $PS_NODES; do
  ssh ${APPSOSUSER}@$node /home/oracle/FSFO/startPS.sh >>$DETAILLOGF &
done
wait

echo ""
echo "`date` -- Start Web Server on All Nodes"
echo "----------------------------------------------------"

for node in $WS_NODES; do
  ssh ${APPSOSUSER}@$node /home/oracle/FSFO/startWS.sh >>$DETAILLOGF &
done
wait

echo ""
echo "----------------------------------------------------"
echo "script completed at `date`"
echo "----------------------------------------------------"
echo
```

The startPS.sh Script

Place the `startPS.sh` script into the same directory on each of the application server nodes.

```
#!/bin/sh

# startPS.bsh is a stand-alone script that starts all domains
# of the PeopleSoft Application Server.

# Set ORACLE_HOME to the client oracle home that the PeopleSoft
application
# server uses.
ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

# Set the correct directory for the following env vars.
export PS_HOME=/u01/app/oracle/product/peoplesoft/PT8.50
export BEA_HOME=/u01/app/oracle/product/peoplesoft/bea
export TUXDIR=/u01/app/oracle/product/peoplesoft/bea/tuxedo10gR3
export COBDIR=/opt/microfocus5/SX50_WP4 ;
export PS_JRE=${PS_HOME}/jre
export ORACLE_DB=PSFT
. /u01/app/oracle/product/peoplesoft/bea/tuxedo10gR3/tux.env ;
. /u01/app/oracle/product/peoplesoft/PT8.50/psconfig.sh ;

# List the domains separated by space i.e., "domain1 domain2"
PS_DOMAINS="PSFT"
WS_DOMAINS="peoplesoft"

LOGF=/home/oracle/FSFO/startPS.log

export ORACLE_HOME PS_HOME
```

```
echo ""  >>$LOGF
echo "----------------------------------------------------" >>$LOGF
echo "script started at `date`"  >>$LOGF
echo "----------------------------------------------------" >>$LOGF
echo ""  >>$LOGF

for domain in $PS_DOMAINS; do
   echo "`date` Starting app server for domain: $domain" >>$LOGF
   ${PS_HOME}/appserv/psadmin -c boot -d $domain >>$LOGF
   echo "`date` Starting Process Scheduler for domain: $domain" >>$LOGF
   $PS_HOME/appserv/psadmin -p start -d $domain >>$LOGF
done

echo ""  >>$LOGF
echo "----------------------------------------------------"  >>$LOGF
echo "script completed at `date`"  >>$LOGF
echo "----------------------------------------------------"  >>$LOGF
echo ""  >>$LOGF
```

The startWS.sh Script

Place the `startWS.sh` script into the same directory on each of the Web server nodes.

```
#!/bin/sh

# startWS.sh is a stand-alone script that starts all domains
# of the PeopleSoft Web Server.

# Set ORACLE_HOME to the client oracle home that the PeopleSoft
application
# server uses.
ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

# Set the correct directory for PS_HOME
PS_HOME=/u01/app/oracle/product/peoplesoft/PT8.50

# List the domains separated by space i.e., "domain1 domain2"
PS_DOMAINS="PSFT"

# WS_DOMAINS is really a list of WLS websites that maps to
# each PeopleSoft app server domains.
WS_DOMAINS="peoplesoft"

LOGF=/home/oracle/FSFO/startWS.log

export ORACLE_HOME PS_HOME

echo ""  >>$LOGF
echo "----------------------------------------------------" >>$LOGF
echo "script started at `date`"  >>$LOGF
echo "----------------------------------------------------" >>$LOGF
echo ""  >>$LOGF
```

```
# Start the Web server for each domain

for domain in $WS_DOMAINS; do
   echo "`date` Starting Web Server for domain: $domain" >>$LOGF
   $PS_HOME/webserv/${domain}/bin/startPIA.sh >>$LOGF
done

echo ""  >>$LOGF
echo "----------------------------------------------------"  >>$LOGF
echo "script completed at `date`"  >>$LOGF
echo "----------------------------------------------------"  >>$LOGF
echo ""  >>$LOGF
```

Appendix C: Client Failover for PeopleSoft Release 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.

> **Note:** Unless indicated otherwise, all of the following steps are applicable to configuring a service for Active Data Guard on Oracle Database 11*g* release 1 (11.1).

Perform the following steps to configure PeopleSoft for client failover for Oracle Database 11*g* release 1 (11.1).

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 "PSFT1,PSFT2" -P BASIC

srvctl add service -d PSFT -s BATCH -r "PSFT1,PSFT2" -P BASIC
```

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

- Both services are defined on the PSFT database.
- Both services run on Oracle RAC instances PSFT1 and PSFT2.
- 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 settings for client failover.

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 services 'HCM' and 'BATCH' 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');
   DBMS_SERVICE.START_SERVICE('BATCH');
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;
```

Note: If you are running PeopleTools 8.51 that is configured with a service for accessing an Active Data Guard standby, this trigger should start the PSQUERY service as well. It should be started on the primary database after (post) failover.

Under normal operations, you will need to create a separate trigger that fires and starts the service on an Active Data Guard standby database when its instance is started. For example:

```
CREATE OR REPLACE TRIGGER manage_ADGservice
after startup on database
DECLARE
role VARCHAR(30);
BEGIN
SELECT DATABASE_ROLE INTO role FROM V$DATABASE;
IF role = 'PHYSICAL STANDBY' THEN
   DBMS_SERVICE.START_SERVICE('PSQUERY');
```

```
END IF;
END;
```

Create this trigger on the primary database, which will then propagate to the physical standby database.

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

After creating the services using the Oracle Clusterware SRVCTL 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 ID 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 ID 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, during which the `TNSNAMES.ORA` connect alias address list is referenced and a connection is established 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 ID 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 a 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. This is described in the "Automating Management Tasks with Data Guard Broker" section earlier in this white paper and in Appendix B.

References

1. Oracle Maximum Availability Architecture Web site
   http://www.otn.oracle.com/goto/maa

2. "Reducing PeopleSoft Downtime Using a Local Standby Database" white paper
   http://www.otn.oracle.com/goto/maa

3. *Oracle Database High Availability Overview (Part #B14210)*
   http://www.oracle.com/pls/db112/lookup?id=HAOVW

4. *Oracle Database High Availability Best Practices (Part B25159)*
   http://www.oracle.com/pls/db111/lookup?id=HABPT

5. *Oracle Real Application Clusters Administration and Deployment Guide*
   http://www.oracle.com/pls/db112/lookup?id=RACAD

6. *Oracle Clusterware Administration and Deployment Guide*
   http://www.oracle.com/pls/db112/lookup?id=CWADD

7. *Oracle Data Guard Concepts and Administration*
   http://www.oracle.com/pls/db112/lookup?id=SBYDB

8. *Oracle Data Guard Broker*
   http://www.oracle.com/pls/db112/lookup?id=DGBKR

9. "Oracle Data Guard Fast-Start Failover" white paper
   http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_FastStartFailoverBestPractices.pdf

10. *PeopleTools 8.51 PeopleBook: Data Management*
    http://download.oracle.com/docs/cd/E18083_01/pt851pbr0/eng/psbooks/tadm/book.htm

Deploying Oracle PeopleSoft Maximum Availability Architecture
February 2011
Author: Darryl Presley
Contributing Authors: Ravi Shankar, Suhas Kulkarni, Jerry Zarate, Linda Koose, Bernice Eng, Lawrence To, Mike Smith, Nancy Ikeda Viv Schupmann

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