Engineering White Paper

EMC²
where information lives

No Data Loss with Oracle8i Data Guard and SRDF

Benefits of Combining EMC Symmetrix Remote Data Facility (SRDF) with Oracle8i Data Guard

Date 4/22/2002

Part Number H429.1

Printed 4/22/2002

Table of Contents

Introduction

Today, IT managers face the problem of choosing the best high-availability (HA) solution at a reasonable cost. Both Oracle and EMC have products that are designed to help limit planned and unplanned outages, providing a flexible HA solution while keeping costs at a minimum.

Oracle8*i* Data Guard™ and EMC Symmetrix Remote Data Facility (SRDF™) complement each other by offering an integrated, high-availability, no-data-loss solution. This solution offers less complexity at a lower cost compared to a full database mirroring solution or an Oracle8*i* Standby Database solution without Data Guard.

Some Oracle customers have been using the EMC SRDF product to remotely mirror the entire database to a failover site. If the primary site failed, everything would be available at the remote location. Oracle has already validated this approach when all volumes are mirrored synchronously. However, heavy update volumes result in a significant amount of communication between sites resulting in higher costs for Telco lines and higher write I/O latency as distances increase.

For a list of vendors that have been validated, refer to the *Oracle Storage Compatibility Program* on the Oracle website.

EMC and Oracle have jointly tested and validated Oracle8*i* Data Guard with EMC SRDF technology. Oracle8*i* Data Guard automates and monitors many of the manual operations that are needed to maintain a standby database, and significantly improves the automation and monitoring of standby databases compared to the standard Standby Database features of Oracle8*i*.

Using EMC SRDF technology, you can easily switch database roles for maintenance purposes or failover to the standby site in case of the primary site failure, without loss of committed transaction data. With SRDF technology, Data Guard switchover is faster, and full recovery of committed transactions is possible even when the primary site is not accessible, this provides for an optimum disaster recovery and data protection solution.

EMC technology can also be used as a no-data-loss solution in a standard Oracle8*i* Standby Database environment. This solution is documented in the EMC white paper, *No Data Loss Standby Database*, which also describes how to create a Standby Database with SRDF.

Purpose

This white paper documents and validates the following features and shows the high level of integration between EMC SRDF and Oracle8*i* Data Guard:

- Switching roles of the primary and standby database
- No-data-loss failover in the case of primary site disaster
- After Data Guard failover, rebuilding the Standby Database at the primary site with SRDF technology or from backup

The Standby Database and SRDF combination was fully tested as part of this project, which requires only that the online redo logs be synchronously mirrored. Such mirroring drastically reduces the volume of data that needs to be transmitted, making this a disaster recovery solution that is applicable to a much broader market.

This solution does not provide protection against logical errors, I/O faults, or similar references that would get propagated using the redo logs.

Standby databases can be used to protect against logical errors by lagging the Standby Database update cycle to be that of the primary database. Standby databases do not recover any work that was done in *unrecoverable* or *nologging* mode. However, Oracle8*i* Data Guard provides a set of monitors that help detect and minimize these cases.

EMC SRDF Overview

SRDF is a Symmetrix®-based business continuance and disaster recovery solution sold as a separate license by EMC Corporation. In simplest terms, SRDF is a configuration of multiple Symmetrix units that maintain multiple, real-time copies of logical volume data in more than one location (Figure 1). SRDF duplicates production (source) site data to a recovery (target) site transparently to users, applications, databases, and host processors. If the primary site is no longer able to continue processing, data at the secondary site is current up to the last I/O transaction.

SRDF can be used in several key areas including:

- Disaster recovery
- Remote backup
- Data center migration
- Symmetrix Data Migration Service (SDMS)
- Data center decision solutions

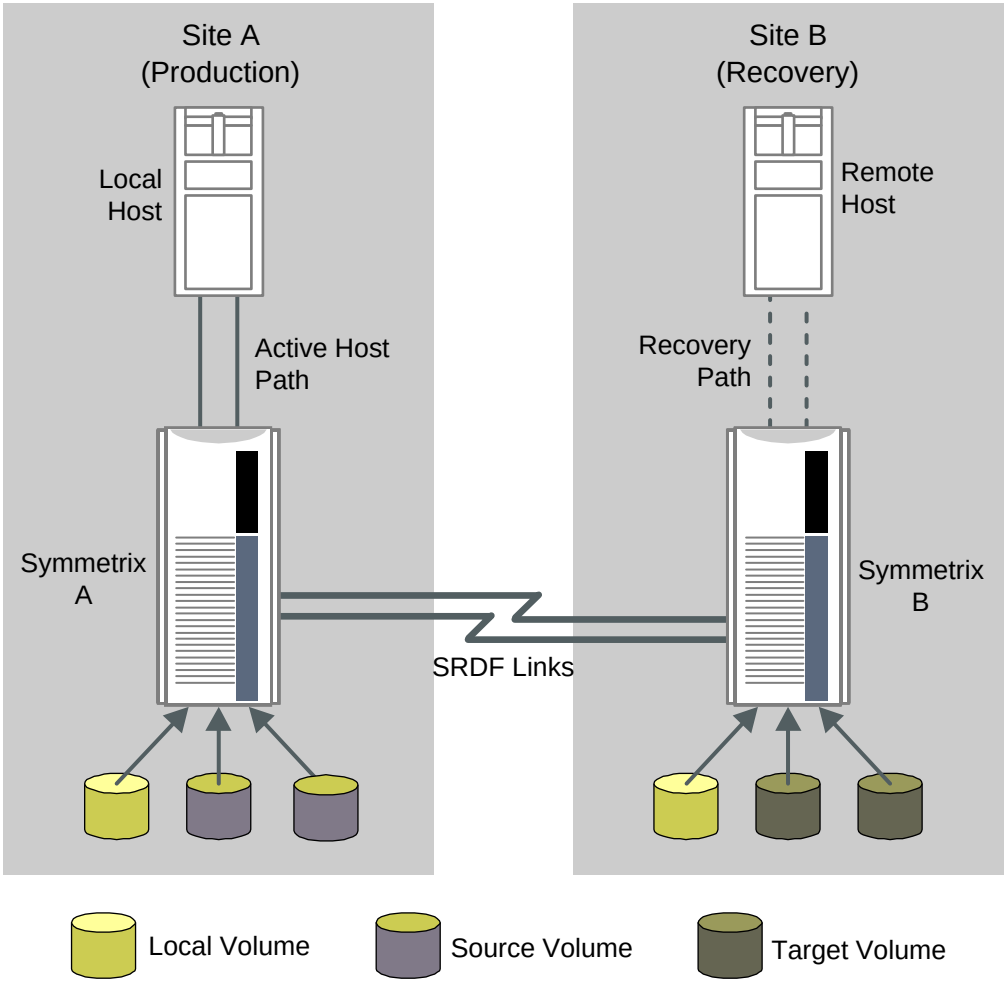

Figure 1. Basic SRDF Configuration

When primary (source) systems are down, SRDF enables fast switchover to the target copy of the data so that critical information is again available in minutes. Business operations and related applications may resume full functionality with minimal interruption. By protecting the data from loss, the operations and applications can be resumed at the secondary site. SRDF can be used as follows:

- By itself, data processing can be resumed by powering up a standby system and manually restarting.
- In combination with more sophisticated software like Oracle standby database.

Oracle8i Data Guard Overview

Oracle8*i* Data Guard provides a complete, pre-tested Standby Database solution for Oracle8*i* databases. It includes all the software required to configure and run a standby database, and provides automation, monitoring, and control that enhances the Oracle8*i* Standby Database in the following ways:

- Simpler disaster recovery
- Easier scheduled maintenance using the Standby Database
- Implementation of safeguards and automation to protect against operator error and data corruption

Previously, customers had to create their own scripts and tools to manage their database environment and to execute switchovers and failovers. Oracle8*i* Data Guard makes those scripts unnecessary and automates the monitoring and management of the Oracle8*i* standby database. Before performing scheduled maintenance on the production node, you can switch the production role to the Standby Database with a single Oracle8*i* Data Guard command.

Oracle8*i* Data Guard automates the steps for switching over to the standby site. When repair or maintenance work on the original primary site has been completed, Oracle8*i* Data Guard automatically switches back to the original configuration.

You can use Oracle8*i* Data Guard to fail over to the standby site in the following situations:

- A user error or data corruption destroys data on the production node.
- The production node or database is inaccessible.

Features

The following features of Oracle8*i* Data Guard enhance the Oracle8*i* standby database:

- Automation and simplification of complex Standby Database operations
- Switchover and switchback for scheduled maintenance
- Failover for unscheduled outages
- Automatic handling of error conditions
- Built-in monitoring tools to detect faults at production and standby databases
- Easy installation and configuration
- Simple user interface
- Customized archiving options such as providing alternate archive destinations when the default archive destination reaches a user-configurable threshold
- Configurable lag between the standby and primary database to prevent the spread of data corruption to the standby database

The primary site consists of the primary database and the Oracle8*i* Data Guard's production agent that contains the monitors and shipper service. The standby site consists of the Standby Database and an Oracle8*i* Data Guard standby agent that contains the monitors and the applier or recovery agent. The Oracle8*i* Data Guard broker coordinates failover, switchover, and other transitions that occur between the production and the standby sites. Its purpose is to ensure the correct functioning of the production and Standby Database environments. For more detailed information, refer to Oracle8*i* Data Guard documentation.

Oracle8*i* Data Guard Switchover

Switchover enables the production and standby databases to switch roles without having to re-create the standby database. When switchover is complete, the new primary database is open on the standby site, and the new Standby Database is mounted on the primary site.

Switchover recovers the new primary database completely without resetting the online redo logs. A new Standby Database is mounted without being re-created. Subsequent switchovers can be implemented to return the databases to their original sites. Oracle8*i* Data Guard provides a single command that replaces 13 switchover steps that are documented in Oracle white papers. Furthermore, integrating with EMC SRDF dramatically reduces Data Guard switchover completion time.

For scheduled maintenance, a user can use the Oracle8*i* Data Guard command line to execute the switchover and stop the standby agent. The user can then upgrade or repair the hardware. Switchover can also be used to test new hardware or to switch hardware in order to diagnose performance problems on the production host.

Figure 2 illustrates the switchover process. During normal operation, the primary database is active, and the archived redo logs are applied to the standby database. After switchover, the primary database is active at the former standby site, and the Standby Database is receiving archived redo logs at the former primary site. Role reversal has occurred without re-creating the standby database. Role reversal refers to the Standby Database residing on the former primary site and the primary database residing on the former standby site.

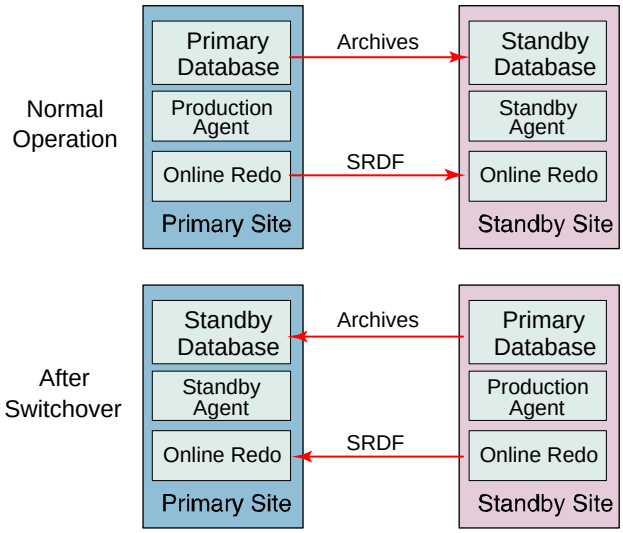

Figure 2. Switchover to the Standby Database

Oracle8*i* Data Guard Failover

When the primary database is inaccessible, you can perform a *failover* from the primary database to the standby database. Performing failover to the Standby Database requires activating the standby database.

The Standby Database is recovered completely (with no data loss), recovered to a point in time, recovered to an SCN (system change number), or recovered to an archive sequence number. It is then activated with reset logs as the new primary database.

After you activate the standby database, it ceases to be a Standby Database and becomes a fully functional primary database with a different incarnation. Activating a Standby Database is a *permanent* operation. You cannot undo the activation and return the database to its former role as a standby database. The old archived redo logs, backups, and previous primary database are not compatible with the new primary database. New archived redo logs from the new production cannot be applied to the old datasets or databases with older incarnations.

Oracle8*i* Data Guard provides a single command that replaces the eight complex steps described in previous failover procedures. The final result is the same: the Standby Database becomes the new primary database, and you must create a new standby database. Integrating EMC SRDF, Data Guard failover can provide no-data-loss failover when the primary site is inaccessible, and in general, failover completion time is reduced dramatically.

Figure 3 illustrates the failover process. During normal operation, the primary database is active, and the archived redo logs are applied to the standby database. Next, failover occurs from the primary database to the standby site. Lastly, the new database is operating at the former standby site with a Standby Database at the former primary site. Data Guard must be initialized and deployed at both sites for the agents to be started again.

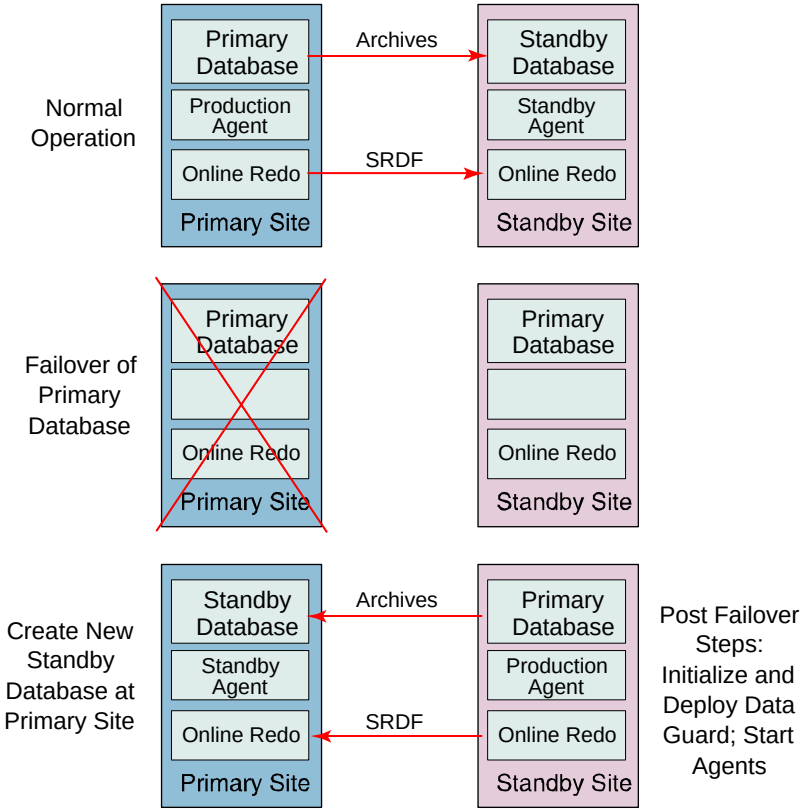

Figure 3. Failover to the Standby Database

Integrating Oracle8i Data Guard with EMC SRDF

Oracle8*i* Data Guard with EMC SRDF minimizes the administrative tasks that are needed to execute a switchover or failover in a Standby Database environment. Integrating Oracle8*i* Data Guard with EMC SRDF offers the following advantages:

- Faster switchover
- A single switchover command for both Oracle and EMC role-reversal commands

  Note: For full protection during an Oracle Data Guard switchover procedure, you can do role reversal of the R1 and R2 (swap) SRDF devices. This allows for protection of the online redo logs in case of a disaster after the Data Guard switchover procedure has successfully completed. If you want this protection, you must be at Symmetrix microcode level 67 and Solutions Enabler 4.2. If you are using an earlier microcode level, contact your local EMC representative and an EMC Customer Engineer can swap roles.

- Faster failover with no data loss of committed transactions

- A single failover command for both Oracle and EMC commands
- Backup solution
- Resilient and faster instantiation and re-instantiation

Switchover and Failover with EMC SRDF Integration

When integrating EMC SRDF technology, the online redo logs are remotely mirrored providing for no-data-loss failover and faster switchover. Oracle8*i* Data Guard has remote mirror callouts that EMC has leveraged to provide an integrated solution for all modes of switchover and failover.

EMC has worked closely with Oracle to provide a callout program that reacts properly for the following Data Guard commands:

- SWITCHOVER
- SWITCHOVER RESTART
- SWITCHOVER ROLLBACK
- FAILOVER
- FAILOVER RESTART
- FAILOVER ROLLBACK

For full data protection of the online redo logs after a Data Guard switchover, EMC SRDF allows swapping roles of the redo device(s) from master to slave. This can be automated into an Oracle8*i* Data Guard application callout script, allowing an integrated solution from a single command. With this functionality, the online redo logs of the new primary database are protected.

Roles of the online redo device(s) can be manually swapped after the Oracle8*i* Data Guard switchover procedure completes. Maintenance on the primary site can continue while the EMC SRDF swap procedure is manually executed.

During the time after the Data Guard switchover procedure complete sand until the SRDF role reversal completes, there is no data protection of the online redo device(s) in case of a disaster.

Installing and Configuring Oracle8*i* Data Guard with EMC SRDF

Installing and configuring Oracle8*i* Data Guard with EMC SRDF consists of three separate operations:

- Downloading and installing Data Guard
- Modifying the Oracle8*i* Data Guard configuration file
- Deploying Oracle8*i* Data Guard

Downloading and Installing Data Guard

1. Log on to the production host as the oracle user or the user that owns the Oracle executables.

2. Download the Oracle8*i* Data Guard software and copy it to the `ORACLE_HOME` directory. If downloading using ftp, you must use binary transfer mode.

3. Change to the `ORACLE_HOME` directory, and then uncompress and untar the software:

```
% cd $ORACLE_HOME
% zcat Oracle8i_DataGuard_R2.6.1_Sun.tar.Z | tar xf –
```

   The tar filename is dependent on the operating system. This example shows installation for a Sun system.

Modifying the Oracle8i Data Guard Configuration File

1. Copy the Data Guard configuration file (`dgd_TEMPLATE.ora`) to the `ORACLE_HOME/dataguard/conf` directory:

```
% export DATAGUARD_ID=app1
% cp $ORACLE_HOME/dataguard/conf/template/dgd_TEMPLATE.ora
$ORACLE_HOME/dataguard/conf/dgd_$DATAGUARD_ID.ora
```

2. Modify mandatory Oracle parameters and review site-specific and optional parameters.

3. Modify the following mandatory EMC parameters to integrate EMC technology:
   - `REMOTE_MIRRORED_LOGFILES=TRUE`
   - `REMOTE_MIRRORED_CALLOUT=TRUE`
   - `REMOTE_MIRRORED_CALLOUT_PROGRAM_HOST1="/full path/Remote_Mirror_Callout_Script"`
   - `REMOTE_MIRRORED_CALLOUT_PROGRAM_HOST2="/full path/Remote_Mirror_Callout_Script"`

Note: Make sure to enter the full path for the `REMOTE_MIRRORED_CALLOUT_PROGRAM_HOST`*n* and that the script is executable.

On both Data Guard switchover and failover, the callout script is executed on the current Standby Database site.

Deploying Data Guard

1. Enter the following commands to distribute the new software to the standby system:

```
% cd $ORACLE_HOME/dataguard/bin
% ./dgdeploy
```

2. Create the new Data Guard database objects (if required).

3. Verify the Data Guard configuration.

To change any Data Guard parameters, issue a `CHANGE` <*Parameter_Name*> <*Value*> command from the Data Guard control program command line (dgdctl).

Using Oracle8*i* Data Guard Switchover with EMC SRDF

With Oracle8*i* Data Guard switchover, you can easily and safely switch the roles of a production and Standby Database for planned maintenance purposes, such as replacing or adding hardware at the primary site.

You can now switch Oracle database roles so that the primary database is on the standby site and the Standby Database is on the primary site. Once the switchover has successfully completed, you issue a Data Guard `STOP STANDBY AGENT` command that shuts down the Standby Database on the primary site and allows you to perform maintenance.

Once the scheduled outage is complete, you issue the `START STANDBY AGENT` command. Optionally, you can then switch back to the original roles.

With SRDF integrated with Data Guard, the switchover does not have to copy the online redo logs since they are already remotely mirrored. For some sites, this is a significant time reduction for switchover. Furthermore, one single Data Guard command issues all the database and EMC commands to provide a safe, simple, and supported switchover procedure. Figure 4 illustrates Data Guard in Normal mode.

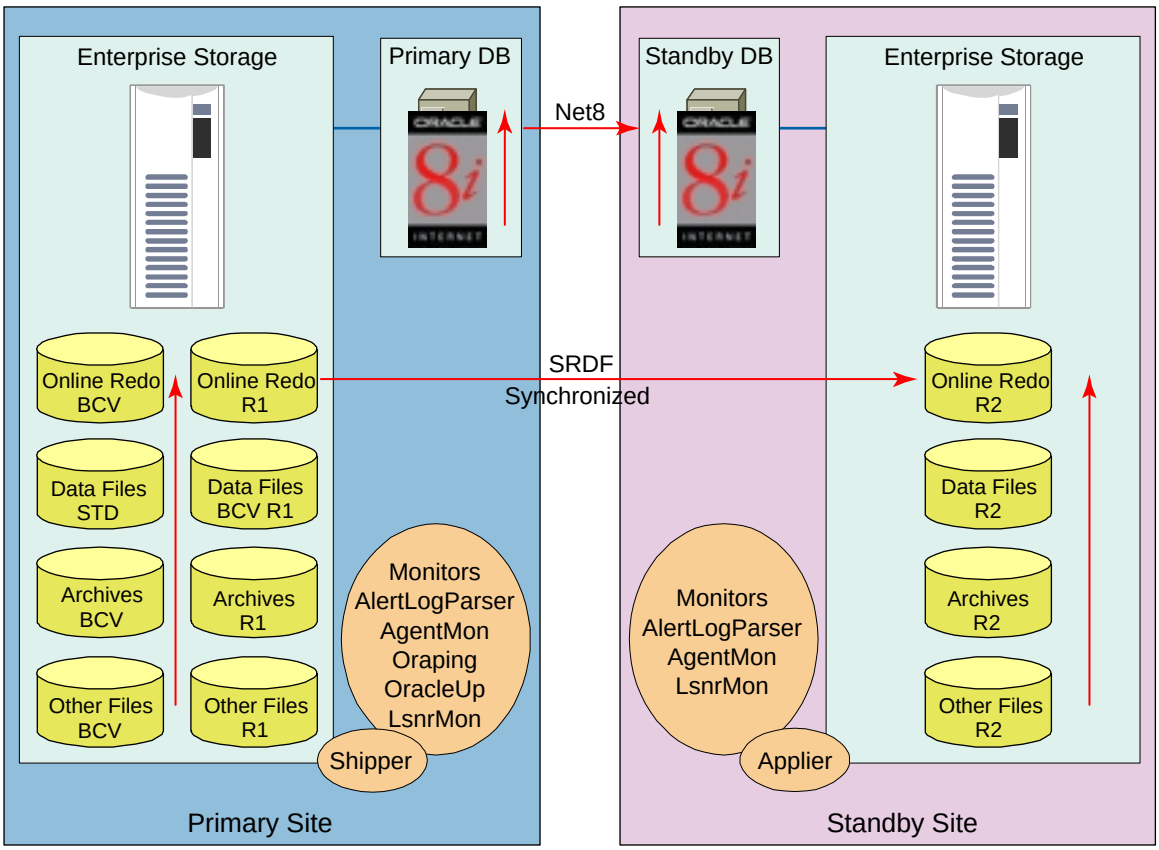

Figure 4. Normal Data Guard Mode

Performing a Data Guard Switchover with SRDF

An Oracle Data Guard switchover procedure includes swapping SRDF device roles (R1 to R2 and R2 to R1). Swapping of SRDF devices can be automated within Data Guard by adding the EMC calls to the switchover callout program and enabling the switchover callout facility within Data Guard.

In the test environment, the swapping of the SRDF redo device group was executed manually after the Data Guard switchover completed. The advantage of the manual approach is that Data Guard switchover can successfully complete and you can continue with the necessary Data Guard commands to stop the standby agents for maintenance purposes. The switchover steps can be de-coupled from the EMC SRDF swap procedure and possibly provide a faster scheduled outage timetable. Customers can automate the SRDF swap procedure into Data Guard's callout script or choose the manual approach.

To perform an Oracle Data Guard switchover, follow the steps below:

1. As `Oracle` user or the user that installed Data Guard, invoke the Data Guard control program by issuing the command, **`dgdctl`**.

2. From the `dgdctl` prompt, issue the **`switchover`** command. The following occurs:

    a. Oracle performs its procedures and eventually uses the Remote Mirror Callout script (refer to step 6 in Table 1). The Remote Mirror Callout arguments dictate how SRDF handles the online redo device group. Since this is a normal switchover procedure, the SRDF template performs an RDF failover, which is unrelated to Data Guard's Failover procedure. The EMC SRDF command will write-disable the R1device(s), write-enable the R2 device(s), and then suspend communication between the SRDF device groups. This allows the standby site to see

the online redo device(s). Depending on the type of volumes on which the online redo logs reside (either Raw or on Filesystem), you need to do the necessary volume management for the Standby host to be able to see those volumes.

b. On the successful completion of the Remote Mirror Callout script (with no Data Guard switchover processing issues), Data Guard prompts for final confirmation (refer to Step 8 in Table 1) to continue with the switchover. Enter your response whether to continue:

- If Yes, then Oracle brings up the primary database on the standby site, brings up the Standby Database on the primary site, and starts all the Data Guard agents.
- If No, then the switchover procedure stops. You can execute a `SWITCHOVER ROLLBACK` or `SWITCHOVER RESTART`. If switchover fails in any step, you should resolve any issues that cause the failure before proceeding with the `ROLLBACK` or `RESTART` option. (Refer to the switchover procedures in Table 1 to verify if a `ROLLBACK` or `RESTART` is possible.)

    - If you perform a SWITCHOVER ROLLBACK and a failure occurs at or after step 6, the SRDF callout script executes. For a rollback, the original production host will need to see those SRDF redo volume(s). The production volumes are write-enabled through the failover or Failback SRDF commands and the necessary volume management.
    - If you perform a SWITCHOVER RESTART and the failure occurred at or prior to step 6, the SRDF callout script executes. The SRDF callout script attempts to write-enable the standby site, write disable the primary site, and perform volume management of the online redo volume(s).

3. The EMC SRDF role reversal (swap) procedure can be *automatically* executed within the Oracle8*i* Data Guard application callout script (put a link to the sample call out script to automate the SRDF swap procedure) or *manually* executed after the Data Guard switchover procedure completes. During the SRDF swap procedure, there is no downtime for the primary database. Follow these steps to complete the EMC SRDF role reversal.

    a. On the standby site (where the SRDF R2 devices resides), execute the following command as user root:

    ```
    symrdf –g <redo device group> swap –refresh R1
    ```

    b. Upon completion, issue the command:

    ```
    symrdf –g <redo device group> est
    ```

    c. Once the SRDF redo device group is synchronized, the customer's online redo logs are protected and the customer can do a full recovery in case of a disaster.

    To verify that the new primary database is functional, queries and updates were run to verify existing data. Furthermore, the new Standby Database was also verified by generating more redo logs and ensuring that the archived logs were being shipped and applied to the new standby database. Figure 5 ilustrates Data Guard after switchover.

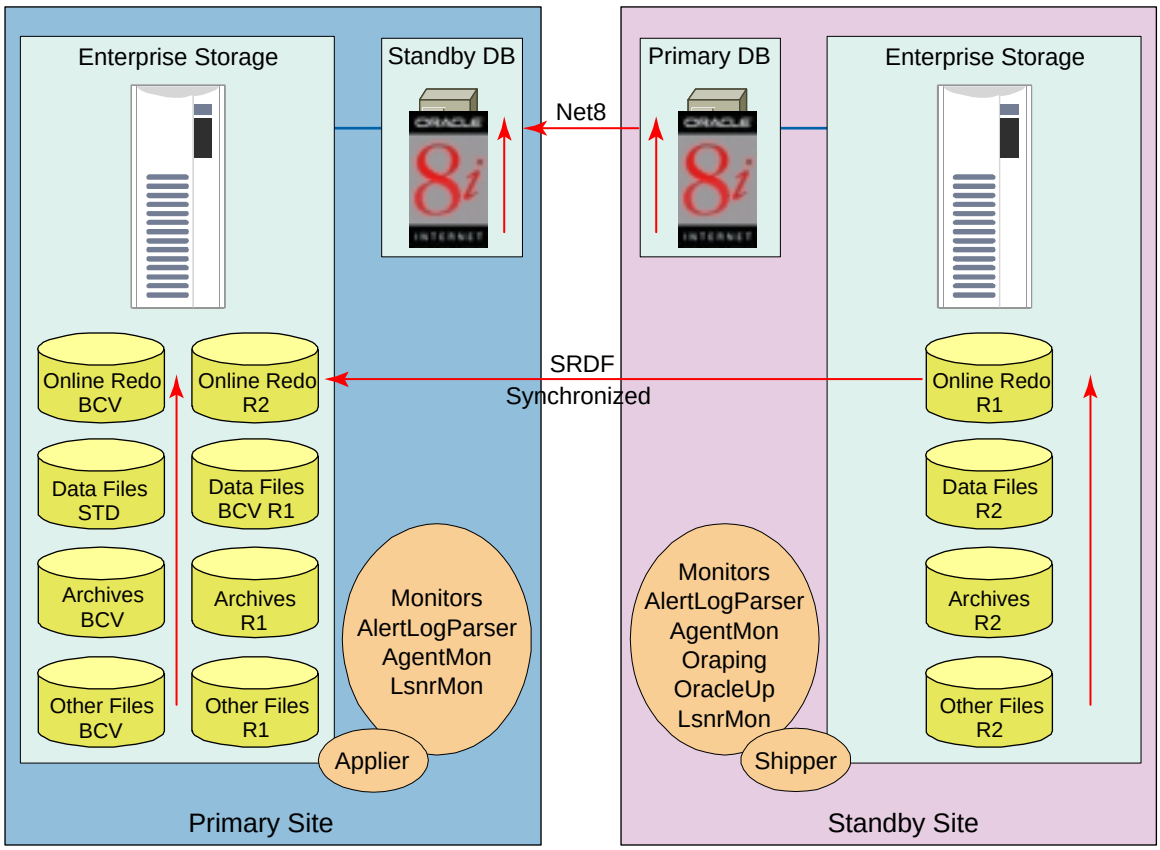

Figure 5. Data Guard after Switchover

Switchover Procedures: Restart and Rollback Options

Table 1 describes the steps in the switchover procedure, and indicates if restart or rollback is possible on each particular step. Restart or rollback is only required after a failed step or when switchover prompts the user for a final confirmation before completing switchover or failover.

The EMC remote mirror callout program is executed in switchover step 6.

Table 1. Switchover Process — Restart and Rollback Options

Step	Task	Restart	Rollback
1	Ensure that the environment is ready for switchover.	Steps 1-13	Step 1
2	Collect switchover information.	Steps 2-13	Steps 2-1
3	Shut down the primary database and the production agent.	Steps 3-13	Steps 3-1
4	Complete full recovery by applying all archives but not the online redo logs.	Steps 4-13	Steps 4-1
5	Shut down the Standby Database and the standby agent.	Steps 5-13	Steps 5-1
6	Copy the online redo logs, if necessary. EMC's remote mirror callout program is executed instead of Data Guard copying the online redo logs.	Steps 6-13	Steps 6-1
7	Copy the password file if necessary.	Steps 7-13	Steps 7-1

Step	Task	Restart	Rollback
Step	Task	Restart	Rollback
8	Issue final user prompt to proceed. Switchover cannot be rolled back after this step.	Steps 8-13	Steps 8-1
9	Open the new primary database.	Steps 9-13	Not allowed
10	Mount the new standby database.	Steps 10-13	Not allowed
11	Complete post-switchover actions. Resolve any archive log gap between the new production and standby databases.	Steps 11-13	Not allowed
12	Start the production agent.	Steps 12-13	Not allowed
13	Start the standby agent.	Step 13	Not allowed

Oracle8*i* Data Guard Switchback with EMC SRDF

After maintenance changes, you may want to switch back or return to the original state where the primary database resides on the primary site an the Standby Database resides on the standby site. In order to switch back, you repeat the same switchover commands. Switchover and switchback are the same within Data Guard. Note that Data Guard executes the callout programs found on the site that contains the current standby database.

To complete the switchback procedure, follow the steps in the earlier section, *Performing a Data Guard Switchover with SRDF*.

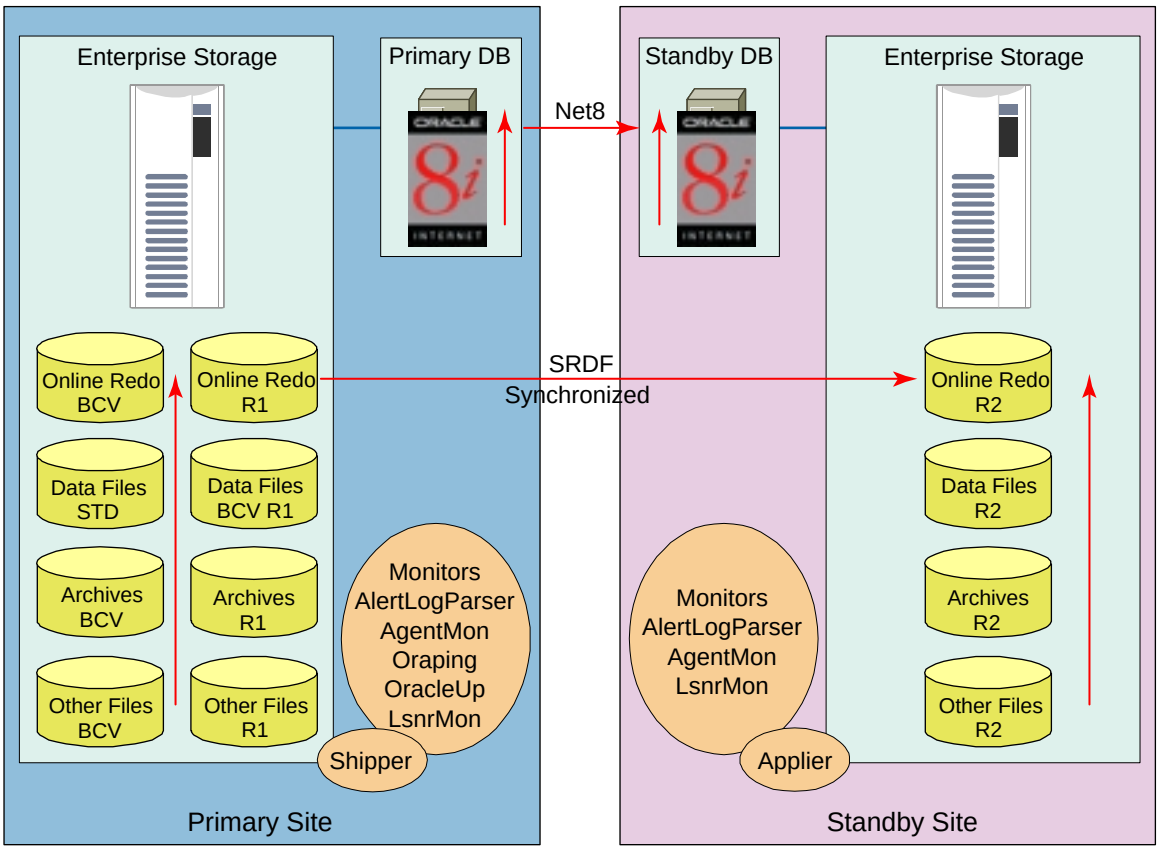

Figure 6. Data Guard Switchback

Oracle8*i* Data Guard Failover with EMC SRDF

In situations where disasters strike on the primary site, Oracle8i Data Guard with EMC SRDF allows the customer to easily and safely activate the Standby Database as a primary database without any loss of data. The primary database will now reside on the standby site, and the users can continue with normal operations while the problem is resolved on the primary site (Figure 7).

When Oracle activates the Standby Database as a primary database, the database is open with the `Resetlogs` operation. The Resetlogs operation creates a new incarnation of the database, which is incompatible with the previous primary database and any prior backups. It is essential to make a backup of the new database as soon as possible.

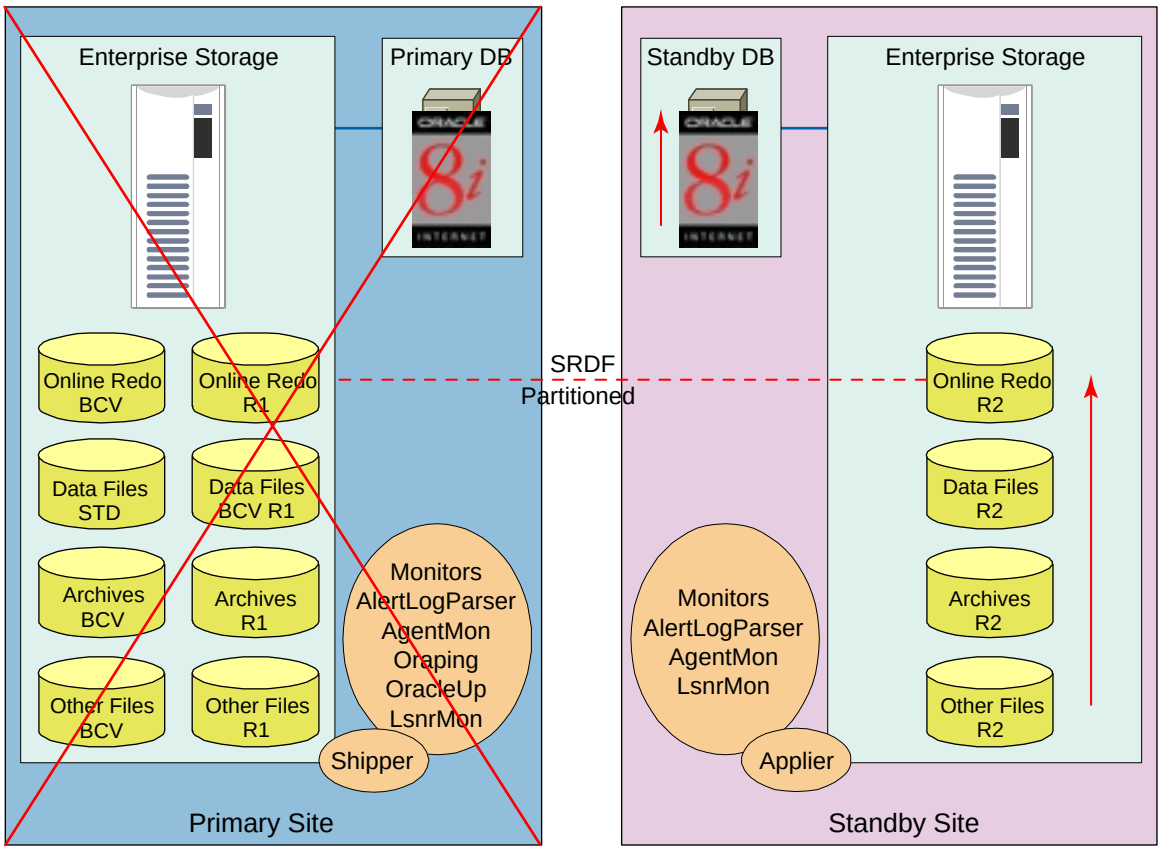

Figure 7. Data Guard Failover

Performing a Data Guard Failover with SRDF

This section describes how to perform the Data Guard failover procedure. After the failover procedure has successfully completed, the standby site contains a new incarnation of the primary database and you can start analyzing the cause of the disaster on the primary site. Customers should make a backup of the new database as soon as possible.

The following are the steps for the Data Guard failover procedure:

1. As the `Oracle` user or the user that installed Data Guard, invoke the Data Guard control program:

   ```
   dgdctl
   ```

2. From the `dgdctl` prompt, issue the **`failover`** command. The following occurs:

    a. Data Guard performs the failover procedure and calls the remote mirror callout script (see Step 3 in Table 2). The remote mirror callout arguments dictate how EMC SRDF handles the online redo device group. Since this is a normal failover procedure, the SRDF script will perform an SRDF failover, which is different from Oracle Data Guard failover procedure. Depending on the state of the RDF link, the RDF script will try to write-disable the R1 device(s), write-enable the R2 device(s), and perform volume management to enable the online redo logs to the standby site. In the case where the callout script cannot write-disable the R1 device(s), the online redo device(s) are in a split-brain mode. The customer should be aware of this and know which data is the valid data before re-establishing the RDF communication between the device(s).

    b. On the successful completion of the remote mirror callout script, Data Guard proceeds with the failover steps in Table 2 until Step 6, the final confirmation. The customer must then decide to continue with the activation of the standby database. Once the customer answers yes, there is no option to roll back Data Guard failover. The Standby Database is opened with Resetlogs as a new incarnation of the database, and all previous backups are incompatible with this new primary database. You should clean out all archive logs of the older incarnation database.

    c. If the customer chooses not to activate the standby database, the Data Guard failover procedure stops. The customer can perform a `FAILOVER RESTART` or `FAILOVER ROLLBACK`. If failover fails due to errors, the customer should resolve any errors that caused the failure before executing the `FAILOVER RESTART` or `FAILOVER ROLLBACK`. Refer to the failover procedures in Table 2 to verify if a `ROLLBACK` or `RESTART` is possible.

        - If the customer chooses to perform a `FAILOVER ROLLBACK` operation and the failure occurs at or after step 3, the remote mirror SRDF callout script will execute. In this case, the original production host must be able to write to the online redo volumes (SRDF redo device group). Depending on the status of the primary site and the RDF link, the customer needs to run the `SRDF` command with option `Failover` or `Failback`.
        - If the customer chooses to perform a `FAILOVER RESTART` and the failure occurs at or before step 3, the remote mirror SRDF callout script will execute. The SRDF callout script will try once again to write-enable the standby site, write-disable the primary site if it can, and perform volume management of the online redo volume(s).

Caution: The Data Guard `Failover` command executes steps 1 through 5 and report errors in step 6. If step 3 failed due to the remote mirror callout and full recovery is intended, you must fix the callout script and issue a `FAILOVER ROLLBACK` and a subsequent `FAILOVER` command to allow for full recovery to complete. In the RDF_Callout_Script provided, the option for manual intervention is given so that the online redo device(s) are made writable to the standby site, so that Data Guard failover can proceed with the option of full recovery without having to issue a `FAILOVER ROLLBACK` and subsequent `FAILOVER`.

3. The customer should run some queries and updates to verify that the new primary database is functional (Figure 8).

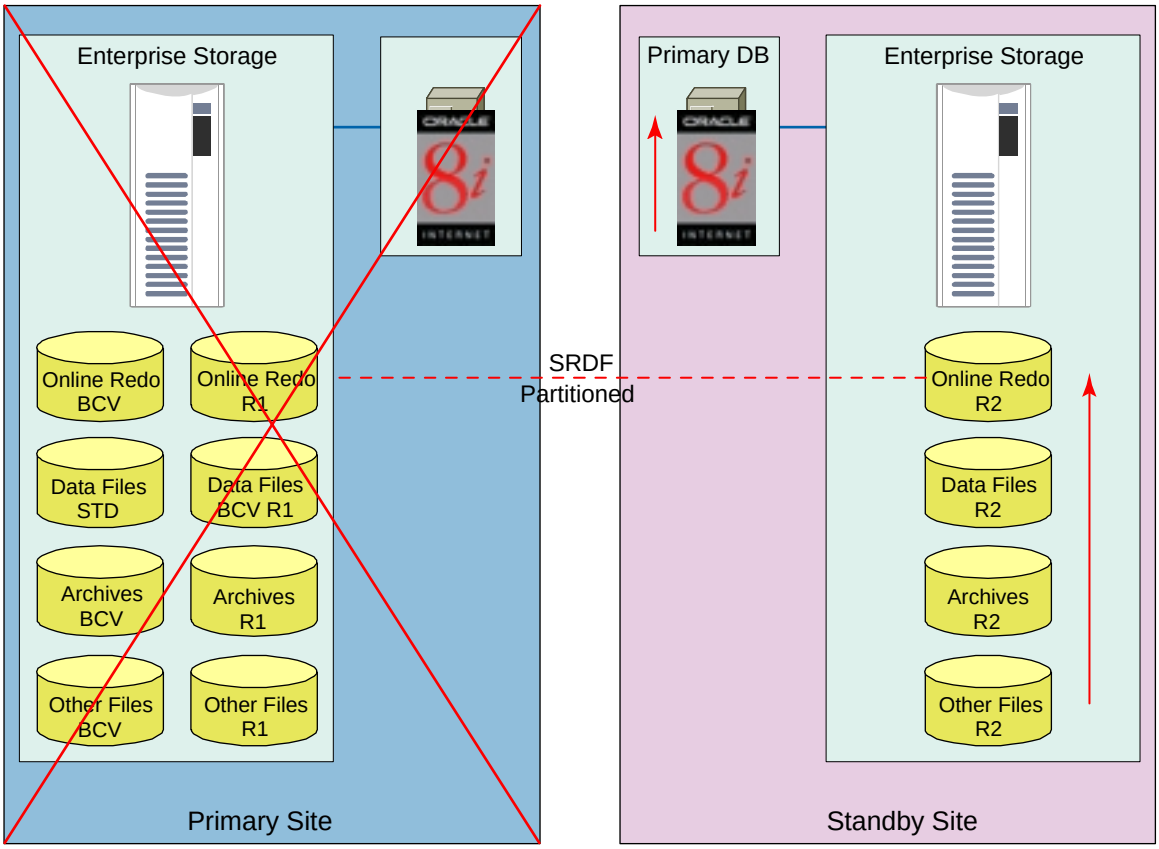

Figure 8: Data Guard after Failover

Failover Procedures: Restart and Rollback Options

Table 2 describes the steps in the failover procedure and indicates if restart or rollback is possible on each particular step. Restart or rollback is only required after a failed step or when failover prompts the user for a final confirmation before completing failover.

The EMC remote mirror callout is executed in failover step 3.

Table 2. Failover Process — Restart and Rollback Options

Step	Task	Restart	Rollback
1	Ensure that the environment is ready for failover.	Steps 1-8	Step 1
2	Collect failover information.	Steps 2-8	Steps 2-1
3	Shut down the primary database and the production agent. The EMC remote mirror callout program is executed after the primary database is shut down.	Steps 3-8	Steps 3-1
4	Prompt the user for the extent of the recovery. Attempt to recover database to the user-requested time, sequence, SCN, or full recovery. Note: If full recovery completes, a rollback is not allowed.	Steps 4-8	Steps 4-1
5	Shut down the Standby Database and the standby agent.	Steps 5-8	Steps 5-1
6	Report all errors discovered in steps 3 through 5 and	Steps 6-8	Steps 6-1

Step	Task	Restart	Rollback
	prompt the user whether to proceed. **Caution**: If no-data-loss or full recovery is requested and errors are reported for step 3, the remote mirror callout program failed or step 4 (full recovery) failed, then the user must correct the problem and issue `FAILOVER ROLLBACK` and then a subsequent `FAILOVER` command.		
7	Activate the new primary database.	Steps 7-8	Not allowed
8	Complete post-failover actions.	Step 8	Not allowed

No Rollback After Recovery

Note that if full recovery completes in Step 4, a rollback is not allowed.

This is a new feature of Data Guard release 2.6.2. The reason is that when failover is executing full recovery, the Data Guard applier process eventually applies the current online redo log. Once the database applies the current log, it knows that it has reached the end of thread for this thread of redo.

Oracle will not apply further archives from this thread since theoretically its recovery is complete. This becomes a problem if you open the primary database after a successful `FAILOVER ROLLBACK` and create more redo logs. The Standby Database will not be able to apply the redo logs and should be re-created. This is not a problem if you restart the failover.

Re-instantiating the Standby Database with EMC and Oracle8i Data Guard

After resolving the issue(s) that caused the primary site failure, the customer can create the Standby Database from the new incarnation of the primary database by using EMC SRDF technology for quicker instantiation of the standby database.

You must perform the following tasks before creating the Standby Database with EMC SRDF technology:

1. Restore `$ORACLE_HOME` from backup of the primary site, if necessary.
2. Create the Symmetrix device groups.
3. Create archive log device(s).
4. Copy database files from the standby site to the primary site using SRDF.
5. Create the standby control file.
6. Initialize and deploy Oracle8*i* Data Guard.

Restoring $ORACLE_HOME

The customer should restore `$ORACLE_HOME` from tape backups, or will need to reinstall the Oracle binaries. For a fresh install of the Oracle binaries, the customer needs to make sure the `init.ora` files and other necessary that are needed for Data Guard (for example, callout scripts) are restored in their proper location(s).

Creating Symmetrix Device Groups on the Primary Site

The customer must issue the following commands to create the Symmetrix device groups for the primary site. If any rebuild of the Symmetrix `bin` file is needed, contact your EMC representative to resolve those issues before issuing these commands.

To create device groups on the primary site for `data_sdg`, `redo_sdg`, `ctrl_sdg`, and `arch_sdg`, enter the following:

```
#symdg -type regular create <data_dev_group>
#symld -g <data_dev_group> -RANGE <symdev start>:<symdev end> addall dev
#symbcv -g <data_dev_group> -RANGE <bcv start>:<bcv end> associateall dev
#symdg -type rdf1 create <redo_device_group>
#symld -g <redo_device_group> add dev <symdev>
#symdg -type rdf1 create <ctrl_device_group>
#symld -g <ctrl_device_group> add dev <symdev>
#symdg -type rdf1 create <arch_device_group>
#symld -g <arch device_group> add dev <symdev>
```

Creating Archive Log Devices

Create archive log destinations to prepare for Oracle8*i* Data Guard deployment. File systems must match the configuration of the archive destinations in the `dgd_$DATAGUARD_ID.ora` file.

Copying Database Files from the Standby Site to the Primary Site with SRDF

Follow these steps to copy the datafiles from the standby site to the primary site:

1. From the primary site as user root, enter:

   ```
   #symrdf –g <data device group> -force update –bcv
   ```

   Depending on the number of invalid tracks between the primary database and the new Standby Database being created, it is beneficial to run multiple SRDF update commands simultaneously to minimize the time for the final SRDF synchronization call. For the final synchronization call, you must put the primary database in "hot" backup mode, run the final SRDF update, split SRDF devices, and the end database backup mode. To minimize the time the database is in hot backup mode, it is advisable to minimize the time it takes to do the final SRDF update command.

2. On the primary site, issue the following TimeFinder command:

   ```
   #symmir –g <data device group> restore –full
   ```

3. Once this command is initiated, issue volume management calls so that the primary site can access the data files.

Creating the Standby Control file

A standby control file is required on the primary site. Follow these steps to create the standby control file:

1. On the primary database, log in as a user with dba privileges on the Oracle database and issue the following command:

   ```
   alter database create standby controlfile as <filename>;
   ```

2. Copy the standby control file that resides on the primary site to the control files on the standby database. To do so, copy the standby control file to the location `/tmp/stbyctl.ctl.`

For example, the location of the control files for the Standby Database in a test environment:

```
/oracle/8.1.7/oradata/ORASID/control01.ctl (control02.ctl and
control03.ctl)
```

For raw volumes, use the `dd` command,

```
dd if=/tmp/stbyctl.ctl of=/oracle/8.1.7/oradata/ORASID/control01.ctl
dd if=/tmp/stbyctl.ctl of=/oracle/8.1.7/oradata/ORASID/control02.ctl
dd if=/tmp/stbyctl.ctl of=/oracle/8.1.7/oradata/ORASID/control03.ctl
```

For `ufs` file systems, use the copy command (`cp`), for example,

```
cp  /tmp/stbyctl /oracle/8.1.7/oradata/ORASID/control01.ctl
/oracle/8.1.7/oradata/ORASID/control02.ctl
=/oracle/8.1.7/oradata/ORASID/control03.ctl
```

Initializing and Deploying Oracle8*i* Data Guard

Once the above procedures are completed, verify that the Data Guard environment and the necessary files such as. callout scripts and `init.ora` files are in their correct locations. Then follow these steps to initialize and run Oracle8i Data Guard:

1. On the standby site (primary database host) invoke the data guard control program, `dgdctl.`

2. At the `dgdctl` prompt, issue the command:

   initialize

3. After the `initialize` command has completed successfully, issue the command:

   deploy

4. After the `deploy` command has completed successfully, start the Data Guard Agents by issuing the command:

   start all

The new Standby Database can be verified and operational by generating more redo logs, ensuring that the archived logs are being shipped and applied to the new standby database. Currently in this state, the online redo logs are not protected in case of disaster. You must execute the SRDF swap procedure to fully protect the database.

Follow these steps to perform an SRDF swap:

1. On the standby site (where the R2 devices resides), execute the following command as user root:

   symrdf –g <*redo device group*> swap –refresh R1

   This command may take a while to complete.

2. Issue the command,

   symrdf –g <*redo device group*> est

   Once the SRDF redo device group is synchronized, the customer's online redo logs are protected and the customer can perform a full recovery in case of disaster.

From this point, an Oracle Data guard switchover procedure returns the environment back to its original state (Figure 9).

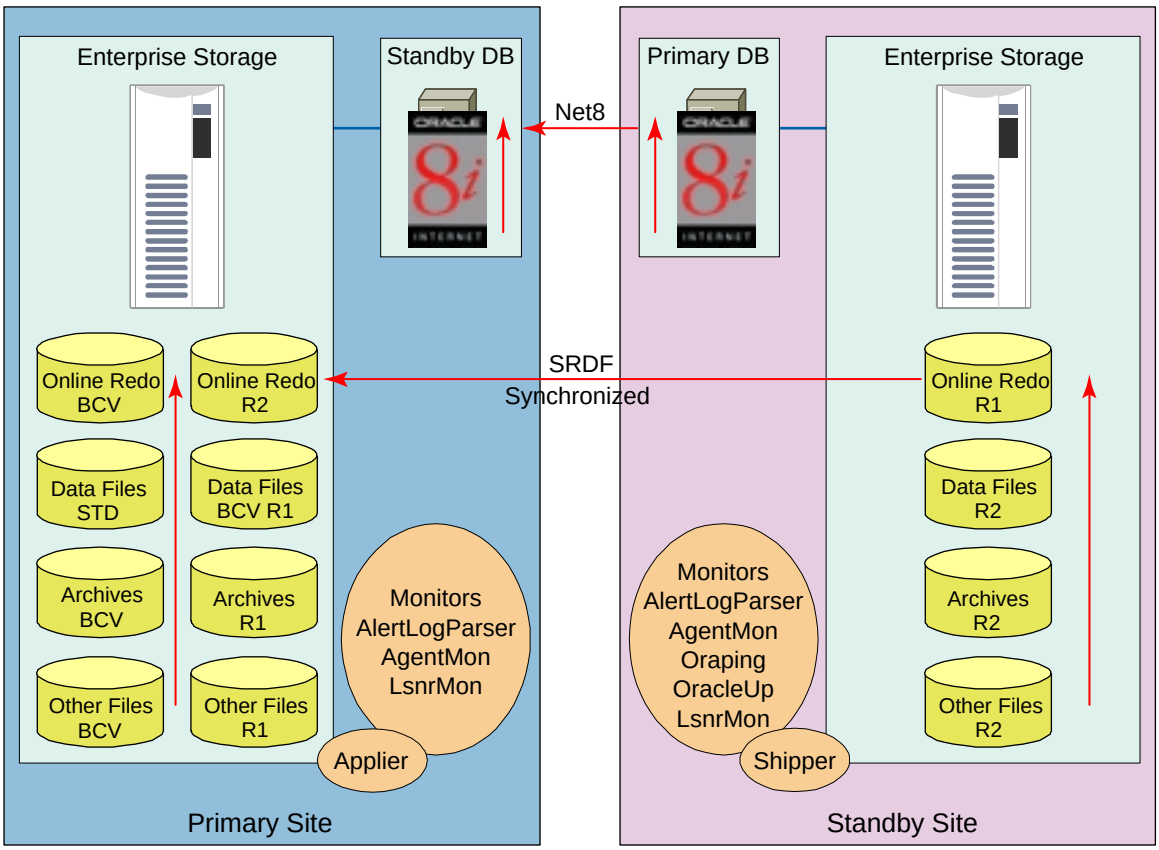

Figure 9. Re-Instantiating Standby Database on the Primary Site

Backup Solutions

You can back up your production data by using the following options:

- EMC TimeFinder with Recovery Manager (RMAN) on the primary database (because of no logging operations)
- EMC TimeFinder or SRDF with RMAN on the standby database
- EMC TimeFinder or SRDF on either database without RMAN
- RMAN alone on either database

EMC recommends using EMC TimeFinder or SRDF with RMAN on the Standby Database when all operations are logged because it offloads CPU usage on the primary database to the Symmetrix unit. Contact your backup vendor for assistance with integrating EMC TimeFinder/SRDF and Oracle RMAN.

Remote Mirror Callout Script Using SRDF

This section provides links to the sample scripts that Oracle provides as a template for the remote mirror callout script. This section also includes a sample template for the EMC SRDF control scripts to integrate with Oracle8*i* Data Guard.

Important: These scripts were validated for the test environment only. You should contact an EMC representative for assistance in changing the scripts to work in their unique environment.

When Remote Mirror Callout is set to true, Oracle8*i* Data Guard passes the following arguments to the callout script in the following order:

- DATAGUARD ID
- Hostname of primary database
- Hostname of standby database
- Transition mode of Data Guard (SWITCHOVER or FAILOVER)
- Mode (NORMAL, RESTART, or ROLLBACK)
- Date

Sample Remote Mirror Callout Script

The sample Remote Mirror Callout script (Oracle8*i* Data Guard template) in Appendix A was used in the test lab environment. With Oracle8*i* Data Guard providing these arguments, the EMC callout can successfully complete Data Guard switchover or failover procedures with varying scenarios (NORMAL, RESTART, or ROLLBACK). With this tight integration, the customer uses one command to handle their scheduled and unscheduled interruptions.

Sample RDF Callout Script

The sample RDF Callout Script (EMC SRDF template for Oracle8i Data Guard) in Appendix B was used to perform the necessary SRDF commands.

Sample Switchover Callout Script

The sample Switchover Callout Script (Oracle8*i* Data Guard template) in Appendix C was used in the test lab environment to automate the role reversal (swap) of the online redo device(s) into the Oracle8*i* Data Guard switchover procedure. This allowed for a single command from Data Guard for a complete role reversal of the databases down to the SRDF device(s), giving data protection upon completion of the switchover procedure.

Note: Volume management examples are not given in the scripts. Once the successful execution of the EMC Solutions Enabler SYMCLI is complete, the customer imports the online redo log disk group, starts the volume(s) on the site where the primary database will reside, and stops the volumes and deports the disk group where the Standby Database will reside.

Summary

Oracle8*i* Data Guard and EMC SRDF create a cost-effective disaster recovery and scheduled maintenance solution that offers an extremely high level of data protection and manageability.

All of the procedures described in this white paper have been validated in the lab, and were successful in both recovering the database on the failover site with no loss of committed transactions, and switching roles between production and standby.

The Oracle8*i* Data Guard and EMC SRDF combination is an effective tool in reducing planned downtime, as it provides quick and smooth switchover from the primary to the standby site and back, without loss of transactions or a reset logs operation. The initial failover, however, does require a resetlogs operation, and subsequently a new Standby Database and backup needs to be instantiated.

As with any implementation of any high-availability solution, a successful implementation requires a solid understanding of all components, a close working relationship of all the parties involved, and extensive validation testing.

Additional Documentation

For Oracle8*i* Data Guard product information, refer to the Oracle Technology Network website.

For more information on SRDF, refer to the SRDF product information.

Refer to the Oracle Metalink to access Oracle8*i* Data Guard documentation and software.

Refer to the *Oracle8i Data Guard Concepts, Administration, and Installation Guide* for more details on Oracle8*i* Data Guard.

Refer to the *Oracle8i Standby Database Concepts and Administration* for more information about Oracle8*i* standby database.

Refer to *No Data Loss Standby Database* white paper on the EMC website.

Refer to *No Data Loss Standby Database* white paper on Oracle's website.

Appendix A: Sample Remote Mirror Callout Script

This is a sample template provided to use with Data Guard switchover and/or failover for REMOTE_MIRRORED_CALLOUT_PROGRAM_HOST*n*. Contact your EMC representative to help you make the necessary changes to this sample for use in your specific environment.

```
#!/bin/ksh
#
# Oracle 8i DataGuard, Release 8.1 - @(#)dgd_remote_mirrored_callout.sh 1.6
2.6.1.0.1
# @(#)dgd_remote_mirrored_callout.sh      @(#)dgd_remote_mirrored_callout.sh  1.6
2.6.1.0.1 - 03/06/01 15:39:21
#
# This is a sample template provided to use with Data Guard switchover and/or
# failover for REMOTE_MIRRORED_CALLOUT_PROGRAM_HOSTn.
#
# Contact your EMC Representative to help you make the necessary changes
# to this sample for use in your specific enviroment.
#
# If the Production online redo log files are being
# mirrored to the Standby system with a Remote Mirroring technology validated
# by the Oracle Storage Compatibility Program (OSCP) (e.g. EMC SRDF in
# synchronous mode), set REMOTE_MIRRORED_LOGFILES=TRUE to skip the transfer of
# the online redo logs (since they are already available on the Standby
# system).  Additionally, the vendor or customer may need to execute specific
# commands to allow the standby system to access these logs.  This callout
# allows for the "vendor or customer" specific commands to occur automatically
# during a failover or switchover.
#
# A "CALLOUT" from failover or switchover will be made
# if the following Data Guard parameters are set as follows (example):
#   REMOTE_MIRRORED_CALLOUT and REMOTE_MIRRORED_CALLOUT_PROGRAM_HOSTn
#   ------------------
#   REMOTE_MIRRORED_LOGFILES=TRUE
#   REMOTE_MIRRORED_CALLOUT=TRUE
#   REMOTE_MIRRORED_CALLOUT_PROGRAM_HOST1=/bin/dgdremotemirrorcallout.sh
#   REMOTE_MIRRORED_CALLOUT_PROGRAM_HOST2=/bin/dgdremotemirrorcallout.sh
#
# If REMOTE_MIRRORED_LOGFILES and REMOTE_MIRRORED_CALLOUT are set to TRUE and
# REMOTE_MIRRORED_CALLOUT_PROGRAM_HOSTn points to a valid program or
# shell script, the script will be run in Switchover Step 6 (Copy Online
# logs) or Failover Step 3 if and only if production database is shutdown
# successfully or it is not accessible.
#
# Thus, the _CALLOUT parms. must be set to TRUE and the PROGRAM parms must be
# set to the full path of the program/shell script.
#
# Customers typically have "vendor specific" commands to write enable the online
# redo logs from the source to the target database for switchover and failover.
#
# Starting with Data Guard 2.6.1 (refer to enhancement bug 1618592) ,
# Data Guard will check and react to the return codes of
# the REMOTE_MIRRORED_CALLOUT_PROGRAM_HOSTn.
# Please refer to the following table for an overview.
#  Callout        Transition    Mode      Action taken w/ non-zero return code
```

```
#  ------------------------------------------------------------------------
#  REMOTE_MIRRORED_CALLOUT
#                 Switchover    NORMAL    switchover fail
#                 Switchover    RESTART   switchover fail
#                 Switchover    ROLLBACK  rollback fail
#                 Failover      NORMAL    prompt to continue failover
#                 Failover      RESTART   prompt to continue failover
#                 Failover      ROLLBACK  rollback fail
#
# For REMOTE_MIRROR_CALLOUTS and switchover, switchover will fail if the remote
# mirror callout fails (returns 1) because the volumes for the online redo logs
# are probably not write enabled on the correct host.  Similarly, SWITCHOVER
# ROLLBACK will fail as well if during a rollback the remote mirror callout
# returns 1.   In both cases, switchover will abort and log the appropriate
# alert messages.
#
# For REMOTE_MIRROR_CALLOUTS and failover, failover will not be able to apply
# any online redo logs if the remote mirror callout fails.  The customer will
# have a choice to continue with incomplete recovery if he desires.  If he
# likes to correct the remote mirror callout, he needs to first correct
# the remote mirror callout problem, issue FAILOVER ROLLBACK and then
# issue FAILOVER again.    During FAILOVER ROLLBACK, a remote mirror
# callout return code of 1 will abort the rollback, log the appropriate
# alert messages and exit.   The customer will be required to fix
# the callout and re-issue the FAILOVER ROLLBACK command.

# Suggestion: Contact the remote mirroring vendor if they have a more specific
# remote mirror callout template specific for Oracle8i Data Guard or refer to
# the http://technet.oracle.com website for the latest templates.

#
# The following arguments are passed to the script in the case listed
#
# DATAGUARD_ID=$1      DATAGUARD_ID of this Data Guard environment
# P_NODE_NAME=$2       hostname of production
# S_NODE_NAME=$3       hostname of standby
# T_MODE=$4            transition mode: SWITCHOVER or FAILOVER
# Mode=$5              NORMAL, RESTART or ROLLBACK
# Date=$6              date mainly for logging/reporting, mm/dd/yy HH:MM:SS
#                      e.g. if a 'failover rollback' was executed then=ROLLBACK
#
# Caution: Be aware that both switchover and failover have 3 modes:
#          NORMAL, RESTART, and ROLLBACK.
#          NORMAL and RESTART commands most probably will be the
#          same but ROLLBACK needs to undo previous commands and allow
#          the initial production database access to the online redo logs.

SUCCESS=0
FAIL=1
so_dgid=$1
p_node=$2
s_node=$3
t_mode=$4
mode=$5
so_date=$6

# Location of callout scripts
```

```
DGPATH=$ORACLE_HOME/dataguard/scripts
SymmRedo=redo_sdg

# The SymmRedo device group is the grouping of 1 or more Symmetrix volumes.
# This enables the Symcli to execute the command(s) to that set of device(s)
# instead of running the cli to each Symmetrix devices.
# The hostname should be captured to validate which host the script was
# initiated on.

currentHost=`hostname`
    if [[ $currentHost == $p_node ]] then
         so_role=PRODUCTION
    elif [[ $currentHost == $s_node ]] then
         so_role=STANDBY
    else
         print -- "ERROR: neither host"
         return 1
    fi
#
print "This is a $t_mode operation running in $mode mode, started \
 on the $s_node host at $so_date for DATAGUARD_ID $so_dgid"
#
# Issue callouts for specific roles
# This is just a sample.  This script needs to be updated for
# your remote mirrored callout requirements.
#
case $t_mode in
    SWITCHOVER )
       case $mode in
         "NORMAL" )
             print "NORMAL SWITCHOVER mode starting..."
             case $so_role in
               "PRODUCTION" )
                  print "Making remote mirror callout on PRODUCTION: \
                  $currentHost"
                  # put vendor callout here with appropriate return values
                  #[[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
                ;;
               "STANDBY" )
                  print "Making remote mirror callout on STANDBY: $currentHost"
                  print "Executing SRDF script..."
# This SRDF script needs to pass the parameters Symmetrix Redo Device Group,
# Transition Mode, Mode (NORMAL, RESTART or ROLLBACK) and Production Node
# to be able to complete successfully.

              $DGPATH/rdf_callout.ksh $SymmRedo $t_mode $mode $p_node

                  [[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
                ;;
             esac
          ;;

         "RESTART" )
             print "RESTART SWITCHOVER mode"
             case $so_role in
               "PRODUCTION" )
```

```
                print "Making remote mirror callout on PRODUCTION: \
                      $currentHost"
                # put vendor callout here with appropriate return values
                # [[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
              ;;
             "STANDBY" )
                print "Restarting remote mirror callout on STANDBY: \
                $currentHost"
                print "Executing SRDF script..."
# This SRDF script needs to pass the parameters Symmetrix Redo Device Group,
# Transition Mode and Production Node  to be able to complete successfully.

             $DGPATH/rdf_callout.ksh $SymmRedo $t_mode $mode $p_node

                [[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
              ;;
          esac
       ;;

       "ROLLBACK" )
          print -- "ROLLBACK SWITCHOVER mode"
          case $so_role in
            "PRODUCTION" )
               print "Making remote mirror callout on PRODUCTION: \
               $currentHost"
               # put vendor callout here with appropriate return values
               # [[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
               ;;
            "STANDBY" )
               print "Making remote mirror callout on STANDBY: $currentHost"

             $DGPATH/rdf_callout.ksh $SymmRedo $t_mode $mode $p_node

                [[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
                ;;
          esac
       ;;
    esac
;;

    FAILOVER )
       case $mode in
         "NORMAL" )
          print -- "NORMAL SWITCHOVER mode"
          case $so_role in
            "PRODUCTION" )
               print "Making remote mirror callout on PRODUCTION: \
               $currentHost"
               # put vendor callout here with appropriate return values
               #[[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
             ;;
            "STANDBY" )
               print "Making remote mirror callout on STANDBY: $currentHost"

             $DGPATH/rdf_callout.ksh $SymmRedo $t_mode $mode $p_node

                [[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
```

```
                ;;
            esac
          ;;

        "RESTART" )
            print -- "RESTART SWITCHOVER mode"
            case $so_role in
              "PRODUCTION" )
                 print "Making remote mirror callout on PRODUCTION: \
                 $currentHost"
                 # put vendor callout here with appropriate return values
                 #[[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
                ;;
              "STANDBY" )
                 print "Making remote mirror callout on STANDBY: $currentHost"

               $DGPATH/rdf_callout.ksh $SymmRedo $t_mode $mode $p_node

                 [[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
                ;;
            esac
           ;;

        "ROLLBACK" )
            print -- "ROLLBACK SWITCHOVER mode"
            case $so_role in
              "PRODUCTION" )
                 print "Making remote mirror callout on PRODUCTION: \
                 $currentHost"
                 # put vendor callout here with appropriate return values
                 #[[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
                ;;
              "STANDBY" )
                 print "Making remote mirror callout on STANDBY: $currentHost"

               $DGPATH/rdf_callout.ksh $SymmRedo $t_mode $mode $p_node

                 [[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
                ;;
            esac
          ;;
       esac
    ;;
esac
```

Appendix B: Sample RDF Callout Script

This is a sample template provided to use to take care of Symmetrix Redo Device Group in a Data Guard switchover and failover situation. Contact your EMC representative to help you make the necessary changes to this sample for use in your specific environment.

```
# Script: /oracle/8.1.7/dataguard/scripts/rdf_callout.ksh
# RDF script to take care of Symmetrix Redo Device Group in Data Guard Switchover
and Failover situation.
#
# Contact your EMC Representative to help you make the necessary changes
# to this sample for use in your specific enviroment.
#
# Location of SRDF binaries

RDFPATH=/opt/emc/SYMCLI/V4.2/bin
DGPATH=$ORACLE_HOME/dataguard/scripts

SymmRedo=$1

# The SymmRedo device group is the grouping of 1 or more Symmetrix volumes.  This
enables the Symcli to
# execute the command(s) to that set of device(s) instead of running the cli to
each Symmetrix devices.

T_MODE=$2
MODE=$3
P_NODE=$4
RDFtype=`$RDFPATH/symld -g $SymmRedo list |grep -i type|awk '{print $4}'`
currentHost=`hostname`
SUCCESS=0
FAIL=1

print "$currentHost Redo Device Group has a RDF type of : $RDFtype"
print "Current mode is : $MODE"

# This script is only runned on the "Standby" host, so if the host is of type R2,
then
# you will need to run a RDF "Failover".  Otherwise, you will need to run a
"Failback".

case $T_MODE in
      "SWITCHOVER" )
         case $MODE in
            "NORMAL" | "RESTART" )
                  case $RDFtype in
                      "RDF1" )
                        print "The Standby Host is of type $RDFtype, so we will
need to do a RDF \
                               Failback..."
                        $RDFPATH/symrdf -g $SymmRedo -noprompt -force failback
                        if [ $? = $SUCCESS ]
                        then
                            ## Customer's script to remove volume access from
Production host
```

```
                            ## Put script here:

                            ## Customer's script for volume management to make
visible redo volumes
                            ## to Standby host.
                            ## Put script here:

                            [[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
                        else
                            print "SRDF command failed.  Please check log for
errors."
                            return $FAIL
                        fi
                       ;; ## End of RDF1

                      "RDF2" )
                        print "The Standby Host is on of type RDF2, so we will
need to do a RDF \
                               Failover..."
                        $RDFPATH/symrdf -g $SymmRedo -noprompt -force failover
                        if [ $? = $SUCCESS ]
                        then
                            ## Customer's script to remove volume access from
Production host
                            ## Put script here:

                            ## Customer's script for volume management to make
visible redo volumes
                            ## to Standby host.
                            ## Put script here:

                            [[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
                        else
                            print "SRDF command failed.  Please check log for
errors."
                            return $FAIL
                        fi
                       ;; ## End of RDF2
                  esac ## End of RDF Type case statement

            ;; ## End of (NORMAL or RESTART)

            "ROLLBACK" )
                  RDFSTATE=`$RDFPATH/symrdf -g $SymmRedo query |grep DEV|awk
'{print $8}'`
                  print "Symmetrix Device Group is : $RDFSTATE"

                  case $RDFSTATE in
                       "RW" )
                        print "Production Redo disk(s) are accessible..."
                            ## Customer's script for volume management to make
visible redo volumes
                            ## to Production host.
                            ## Put script here:

                          [[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
                        ;;
```

```
                        "WD" )
                           print "Production DB redo logs are Write Disabled."
                           case $RDFtype in
                              "RDF1" )  ## This means that Production host is of
type RDF2
                                    $RDFPATH/symrdf -g $SymmRedo -noprompt -force
failover
                                    if [ $? = $SUCCESS ]
                                    then
                                    ## Customer's script to remove volume access
from Standby host
                                    ## Put script here:

                                    ## Customer's script for volume management to
make visible redo volumes
                                    ## to Production host.
                                    ## Put script here:

                                                [[ $? = $SUCCESS ]] && return
$SUCCESS || return $FAIL
                                          else
                                                print "SRDF command failed.
Please check log for errors."
                                                return $FAIL
                                          fi
                                    ;; ## End of RDF1 case

                                 "RDF2" )  ## This means that Production host is of
type RDF1
                                    $RDFPATH/symrdf -g $SymmRedo -noprompt -force
failback
                                          if [ $? = $SUCCESS ]
                                          then
                                    ## Customer's script to remove volume access
from Standby host
                                    ## Put script here:

                                    ## Customer's script for volume management to
make visible redo volumes
                                    ## to Production host.
                                    ## Put script here:

                                                [[ $? = $SUCCESS ]] && return
$SUCCESS || return $FAIL
                                          else
                                                print "SRDF command failed.
Please check log for errors."
                                                return $FAIL
                                          fi
                                    ;; ## End of RDF2 case
                              esac ## End of RDFtype case statement
                        ;; ## End of WD case
                  esac ## End of RDFSTATE case statement
            ;; ## End of ROLLBACK
         esac ## End of (NORMAL, RESTART and ROLLBACK) case statement
      ;;  ## End of SWITCHOVER
```

```
        "FAILOVER" )
           case $MODE in
                "NORMAL" | "RESTART" )
                        case $RDFtype in
                           "RDF1" )
                                print "The Standby Host is of type RDF1, so we
will need to do a RDF \
                                       Failback..."
                                $RDFPATH/symrdf ping
                                if [ $? = $SUCCESS ]
                                then
                                    $RDFPATH/symrdf -g $SymmRedo -noprompt -force
failback
                                    if [ $? = $SUCCESS ]
                                    then
                                        /usr/sbin/ping $P_NODE >> /dev/null
                                        if [ $? = $SUCCESS ]
                                        then
                                            ## Customer's script to remove volume
access from Production host
                                  ## Put script here:
                                        else
                                            print "Can not ping $P_NODE.  Deport
of redo_dg failed. \
                                                    Continuing $T_MODE..."
                                        fi
                                        print "RDF Failback was successful."
                                ## Customer's script for volume management to make
visible redo volumes
                                ## to Standby host.
                                ## Put script here:

                                        [[ $? = $SUCCESS ]] && return $SUCCESS ||
return $FAIL
                                    else
                                        print "SRDF command failed.  Please check
log for errors."
                                print "If full recovery is intended, please manually
correct the errors."
                                print "You will need to do an RDF failback command
to make the Standby \n"
                                print "database online redo logs write_enable and
the Production database \n"
                                print "online redo logs write_disable."

                                        print "Once that has completed, volume
management needs to be completed, \n"
                                print "so that the Standby Database can access the
online redo logs for \n"
                                print "full recovery."

                                print "Answer Y if you have corrected the SRDF
failure and have made \n"
                                print "accessible the online redo logs on the
Standby database. Otherwise, \n"
```

```
                                print "answering N, DG will continue without the
full recovery."

                                print "Answer [Y or N]"
                                      while read answer
                                do
                                   case $answer in
                                            "N" | "n" )
                                        print "You have chosen not to
continue.  Now Data Guard will continue \n"
                                        print "but full recovery is not possible,
unless a Failover Rollback \n"
                                        print "is issued and another DG failover is
started."
                                            return $FAIL
                                            ;;

                                            "Y" | "y" )
                                            print "Continuing with FAILOVER
ROLLBACK...and full recovery is possible."
                                        return $SUCCESS
                                            ;;

                                            * )
                                            print "Please type in Y or N."
                                            continue
                                            ;;
                                   esac
                                done
                                fi
                              else
                                  print "RDF link is DOWN!!!"
                                  print "Will WRITE_ENABLE $SymmRedo device
group on Production host \
                                         $P_NODE."
                                  print "Oracle Full Recovery is not possible."
                                  $RDFPATH/symrdf -g $SymmRedo -noprompt -force
rw_enable r1
                                  if [ $? = $SUCCESS ]
                                  then
                                      print "Oracle FAILOVER will continue..."
                                      ## Customer's script for volume management
to make visible redo volumes
                                ## to Standby host.
                                ## Put script here:

                                      return $FAIL
                                  else
                                      print "SRDF command failed.  Please check
log for errors."
                                print "Since RDF link was down, full recovery is not
an option."
                                print "What manually needs to be done is to
write_enable the online redo logs for \n"
                                print "Standby database so that DG failover can
proceed accordingly.  The RDF \n"
```

```
                                print "command is : symrdf -g <redo device group> -
force rw_enable R2

                                        print "Once that has completed, volume
management needs to be completed, \n"
                                print "so that Data Guard and continue with the
failover process and bring up \n"
                                print "the new incarnation of the database."

                                print "Answer Y if you have corrected the SRDF
failure and have made \n"
                                print "accessible the online redo logs on the
Standby database."
                                print "answering N, DG will continue but will fail
when trying to bring up \n"
                                print "the new incarnation of the database."

                                print "Answer [Y or N]"
                                      while read answer
                                do
                                   case $answer in
                                           "N" | "n" )
                                        print "You have chosen not to
continue.  Now Data Guard will continue..."
                                        return $FAIL
                                           ;;

                                           "Y" | "y" )
                                        print "Continuing with FAILOVER..."
                                   return $FAIL
                                           ;;

                                           * )
                                        print "Please type in Y or N."
                                        continue
                                           ;;
                                   esac
                                done
                                fi
                             fi
                          ;; ## End of RDF1 case

                         "RDF2" )
                              print "The Standby Host is on of type RDF2, so we
will need to do a RDF \
                                     Failover..."
                              $RDFPATH/symrdf -g $SymmRedo -noprompt -force
failover
                              if [ $? = $SUCCESS ]
                              then
                                  /usr/sbin/ping $P_NODE >> /dev/null
                                  if [ $? = $SUCCESS ]
                                  then
                                      ## Customer's script to remove volume
access from Production host
                                ## Put script here:
```

```
                                    else
                                        print "Can not ping $P_NODE.  Deport of
redo_dg failed.  \
                                                Continuing $T_MODE..."
                                    fi
                                    print "RDF Failover was successful."
                                    print "Doing volume management..."
                                    ## Customer's script for volume management to
make visible redo volumes
                               ## to Standby host.
                               ## Put script here:

                                    [[ $? = $SUCCESS ]] && return $SUCCESS ||
return $FAIL
                                  else
                                        print "SRDF command failed.  Please check
log for errors."
                                print "If full recovery is intended, please manually
correct the errors."
                                print "You will need to do an RDF failback command
to make the Standby \n"
                                print "database online redo logs write_enable and
the Production database \n"
                                print "online redo logs write_disable."

                                    print "Once that has completed, volume
management needs to be completed, \n"
                                print "so that the Standby Database can access the
online redo logs for \n"
                                print "full recovery."

                                print "Answer Y if you have corrected the SRDF
failure and have made \n"
                                print "accessible the online redo logs on the
Standby database. Otherwise, \n"
                                print "answering N, DG will continue without the
full recovery."

                                print "Answer [Y or N]"
                                      while read answer
                                do
                                   case $answer in
                                            "N" | "n" )
                                       print "You have chosen not to
continue.  Now Data Guard will continue \n"
                                   print "but full recovery is not possible,
unless a Failover Rollback \n"
                                   print "is issued and another DG failover is
started."
                                         return $FAIL
                                            ;;

                                            "Y" | "y" )
                                         print "Continuing with FAILOVER...and
full recovery is possible."
                                   return $SUCCESS
                                            ;;
```

```
                                        * )
                                    print "Please type in Y or N."
                                    continue
                                        ;;
                                esac
                            done
                        fi
                    ;; ## End of RDF2 case
                esac ## End of RDFtype case statement
            ;; ## End of NORMAL or RESTART case

            "ROLLBACK" )
                    case $RDFtype in
                       "RDF1" )
##
## During a rollback from a Data Guard Failover procedure, the final result that
you will want is to Write
## Enable your Production site, which in this case is the R2.  The command in this
scenario you want to do
## is the symrdf "failover"
##
                            print "You are about to rollback the Data Guard
FAILOVER procedure."
                            print "Please make sure that you understand what you
want to do here."
                            print "Before proceeding with Data Guard FAILOVER
ROLLBACK, open a new \
                                   terminal"
                            print "and make visible the REDO device(s) with the
necessary RDF commands \
                                   and volume management tools."
                            print "Also, Write Disable the redo device(s) on the
Standby host."
                            print "\n"
                            print "In this case, the RDF command you want to run
is: "
                            print "        symrdf -g <Symm redo device group> -
force failover"
                            print "\n"
                            print "Then do whatever volume management you need
to allow your Production \
                                   host see the Redo volumes."
                            print "Once that is completed successfully, then you
can continue. \
                                   Otherwise, Data Guard will stop."
                            print "\n"
                            print "If the Production host already sees the REDO
device(s), then just \
                                   continue with FAILOVER ROLLBACK."
                            print "\n"
                            print "Are you sure you want to continue? [Y or N]"
                            while read answer
                               do
                                 case $answer in
                                   "N" | "n" )
```

```
                                        print "You have chosen not to continue.
Now Data Guard will fail."
                                        print "Please remedy whatever needs to be
done before retrying \
                                               Data Guard FAILOVER ROLLBACK"
                                        return $FAIL
                                     ;;

                                     "Y" | "y" )
                                        print "Continuing with FAILOVER
ROLLBACK..."
                                return $SUCCESS
                                     ;;

                                     * )
                                        print "Please type in Y or N."
                                        continue
                                     ;;
                                  esac
                               done
                            ;; ## End of RDF1 case

                           "RDF2" )
##
## During a rollback from a Data Guard Failover procedure, the final result that
you will want is to Write
## Enable your Production site, which in this case is the R1.  The command in this
scenario you want to do
## is the symrdf failback
##
                                print "You are about to rollback the Data Guard
FAILOVER procedure."
                                print "Please make sure that you understand what you
want to do here."
                                print "Before proceeding with Data Guard FAILOVER
ROLLBACK, open a new \
                                       terminal"
                                print "and make visible the REDO device(s) with the
necessary RDF commands \
                                       and volume management tools."
                                print "Also, Write Disable the redo device(s) on the
Standby host."
                                print "\n"
                                print "In this case, the RDF command you want to run
is: "
                                print "        symrdf -g <Symm redo device group> -
force failback"
                                print "\n"
                                print "Then do whatever volume management you need
to allow your Production \
                                       host see the Redo volumes."
                                print "Once that is completed successfully, then you
can continue.  \
                                       Otherwise, Data Guard will fail."
                                print "\n"
                                print "If the Production host already sees the REDO
device(s), then just \
```

```
                                    continue with FAILOVER ROLLBACK."
                                print "\n"
                                print "Are you sure you want to continue? [Y or N]"
                                while read answer
                                   do
                                     case $answer in
                                       "N" | "n" )
                                          print "You have chosen not to continue.
Now Data Guard will fail."
                                          print "Please remedy whatever needs to be
done before retrying \
                                                 Data Guard FAILOVER ROLLBACK"
                                          return $FAIL
                                       ;;

                                       "Y" | "y" )
                                          print "Continuing with FAILOVER
ROLLBACK..."
                                          return $SUCCESS
                                       ;;

                                       "*" )
                                          print "Please type in Y or N."
                                          continue
                                       ;;
                                     esac ## End of answer case statement
                                   done
                               ;; ## End of RDF2 case
                           esac ## End of RDFtype case statement
                     ;; ## End of ROLLBACK case
                esac ## End of NORMAL, RESTART or ROLLBACK case statement
           ;; ## End of Failover case statement
esac ## End of Transition case statement
```

Appendix C: Sample Switchover Callout Script

This is a sample template provided for use with Data Guard switchover for pre-and/or post-custom processing. Contact your EMC representative to help you make the necessary changes to this sample for use in your specific environment.

```
#!/bin/ksh
#
# Oracle 8i DataGuard, Release 8.1 - 2.6.1.0.1
# @(#)dgd_switchover_callout.sh     1.9 2.6.1.0.1 - 03/06/01 15:39:21
#
# This is a sample template provided for use with Data Guard switchover
# for pre and/or post custom processing.
#
# Contact your EMC Representative to help you make the necessary changes
# to this sample for use in your specific enviroment.
#
# A "CALLOUT" from switchover will be made pre and post processing
# if the following Data Guard parameters are set as follows (example):
#   Switchover Callout
#   ------------------
#    SWITCHOVER_CALLOUT=TRUE
#    SWITCHOVER_CALLOUT_PROGRAM_HOST1=/usr/local/bin/dgdcalloutSO.sh
#    SWITCHOVER_CALLOUT_PROGRAM_HOST2=/usr/local/bin/dgdcalloutSO.sh
#
#
# Thus, the _CALLOUT parms. must be set to TRUE and the PROGRAM parms must be
# set to the full path of the program/shell script.
#
# Customers typically have "applications" that they want to stop on
# the current production host before they start to switch over.
# At the end of switchover, they would restart the application on the
# new Production.
#
# The Data Guard switchover program will check the return codes of the
# pre-switchover callout and post-switchover callouts.  The following
# return codes are recognized:
#               0 - SUCCESS
#               1 and non-zero - FAIL
#
# If the pre-switchover callout returns $FAIL (1), the switchover is
# aborted immediately.  If the post-switchover callout returns $FAIL (1)
# then an alert message will be sent to the screen and to the switchover
# log file.  However, switchover steps will have completed successfully.
# For restart and rollback scenarios, any switchover callouts that
# return $FAIL (1) are not fatal but again switchover will log ALERTS
# into the switchover log and to the command line screen.
#
# Hence, Do NOT return $FAIL for non-critical errors since switchover
# will abort during the initial switchover.

# The switchover callout return code checking was implemented
# with DG release 2.6.1.  Refer to enhancement bug 1618592.
# Please refer to the following table for an overview.
```

```
#  Callout        Status        Mode      Action taken w/ non-zero return code
#  ------------------------------------------------------------------------
#  SWITCHOVER_CALLOUT
#                 stop  (pre)   NORMAL    switchover fail
#                 stop  (pre)   RESTART   N/A
#                 stop  (pre)   ROLLBACK  N/A
#                 start (post)  NORMAL    switchover success, log Alert msg
#                 start (post)  RESTART   switchover success, log Alert msg
#                 start (post)  ROLLBACK  rollback success, log Alert msg

#
# The following arguments are passed to the script in the case listed
#
# DATAGUARD_ID=$1      DATAGUARD_ID of this DataGuard environment
# NODE_NAME=$2         hostname of production or standby (see $3)
# ROLE=$3              PRODUCTION or STANDBY
# Status=$4            start (at the end) or stop (at the beginning)
# Date=$5              date mainly for logging/reporting, mm/dd/yy HH:MM:SS
# Mode=$6              NORMAL, RESTART or ROLLBACK
#                      e.g. if a 'failover rollback' was executed then=ROLLBACK
#
#

so_dgid=$1
so_node=$2
so_role=$3
so_status=$4
so_date=$5
so_mode=$6


# The hostname should be captured to validate which host the script was
# initiated on.
#    e.g. if currentHost=so_node && so_role=STANDBY && so_status=start THEN
#            run callout start procedures
currentHost=`hostname`
#
print "This is a $so_status operation running in $so_mode mode, started \
 on the $so_role host $so_node at $so_date for DATAGUARD_ID $so_dgid"
#
# Start or stop on production or standby .
#

RDFPATH=/opt/emc/SYMCLI/V4.2/bin
DGPATH=$ORACLE_HOME/dataguard/scripts
SymmRedo=redo_sdg

# The SymmRedo device group is the grouping of 1 or more Symmetrix volumes.  This
enables the Symcli to
# execute the command(s) to that set of device(s) instead of running the cli to
each Symmetrix devices.

SUCCESS=0
FAIL=1

RDFtype=`$RDFPATH/symld -g $SymmRedo list |grep -i type|awk '{print $4}'`
```

```
case $so_status in
    start )
      case $so_role in
        "PRODUCTION" )
           print "Starting application on production"
           if [[ $currentHost = $so_node ]]
           then
             print "locally"
             #
             # Start application on production , locally
             #
             [[ $? = $FAIL ]] && return $FAIL
           else
             print "remotely"
             #
             # Start application on production , remotely
             #
             [[ $? = $FAIL ]] && return $FAIL
           fi
           return $SUCCESS
           ;;
        "STANDBY" )
           print "Starting application on standby"
           if [[ $currentHost = $so_node ]]
           then
             print "locally"
             #
             # Start application on Standby , locally
             #
             # After completion of starting the applications, swapping
             # of SRDF online redo device(s) [R1 and R2] is executed below
            case $RDFtype in
                "RDF1" )
                  print "Swapping is not necessary."
                  return $SUCCESS
                ;;

                "RDF2" )
                  print "Swapping online redo device(s)..."
                  $RDFPATH/symrdf -g $SymmRedo swap -refresh R1
                  [[ $? = $SUCCESS ]] && return $SUCCESS || return $FAIL
                ;;
            esac
           else
             print "remotely"
             #
             # Start application on Standby , remotely
             #
             [[ $? = $FAIL ]] && return $FAIL
           fi
           return $SUCCESS
           ;;
        esac
        ;;
    stop )
      case $so_role in
        "PRODUCTION" )
```

```
            print "Stopping application on production"
            if [[ $currentHost = $so_node ]]
            then
              print "locally"
              #
              # Stop application on Production , locally
              #
              [[ $? = $FAIL ]] && return $FAIL
            else
              print "remotely"
              #
              # Stop application on Production , remotely
              #
              [[ $? = $FAIL ]] && return $FAIL
            fi
            return $SUCCESS
            ;;
          "STANDBY" )
            print "Stopping application on standby"
            if [[ $currentHost = $so_node ]]
            then
              print "locally"
              #
              # Stop application on Standby , locally
              #
              [[ $? = $FAIL ]] && return $FAIL
            else
              print "remotely"
              #
              # Stop application on Standby , remotely
              #
              [[ $? = $FAIL ]] && return $FAIL
            fi
            return $SUCCESS
            ;;
        esac
          ;;
esac


#
# Ex - 2 nodes node1 and node2 . node1 is current production and node2 is
#        standby
# Switchover is started in node 2 which is the standby.
# custom script is called twice at beginning, once for production and once
# for standby
#
#
```