

An Oracle White Paper
January 2010

Setup Flashback Database on Data Guard Physical Standby Database for SAP Customers

Database Version : 11g Release 2

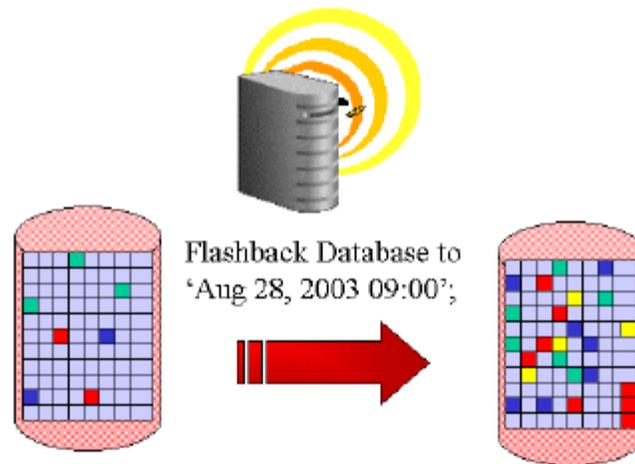
Preface	2
Requirements	4
Aim of the document	4
Setting up flashback logging	5
Resetting the physical standby database to a defined point in time...	9
Using physical standby as primary database after flashback.....	17
Flashback and switchover	19
Flashback and failover	19
Annex	20

Preface

Many studies show that 40% of application outages are caused by operator or user errors. Making mistakes is just part of being human. But these errors are extremely difficult to avoid and can be very difficult to recover from without advance planning and the right technology. Such errors can result in “logical” data corruption or cause one or more components of the IT infrastructure to cease operating. While it is relatively simple to rectify the failure of an individual component, detection of and repairs to logical data corruption, such as accidental deletion of valuable data, is a time-consuming operation that results in a huge loss in business productivity. Typical user errors may include accidental deletion of valuable data, deleting the wrong data, and dropping the wrong table. This paper focuses on the Oracle Database 10g architecture that makes use of the unique technological advances made in database recovery due to human errors – Flashback Technology.

Flashback Database quickly rewinds an Oracle database to a previous point in time to correct any problems caused by logical data corruptions or user errors. Flashback Database is like a ‘rewind button’ for your database.

It provides database point-in-time recovery without the user first having to restore a backup of the database. When you eliminate the time it takes to restore a database backup from tape, database point-in-time recovery is fast. The Flashback Database capability, accessible from both RMAN and SQL*Plus using the FLASHBACK DATABASE command, is similar to conventional point-in-time recovery in its effects. It allows you to return a database to its state at a time in the recent past.



Enabling Flashback Database functionality provides the following benefits.

- *Eliminate the time required to restore a backup.* When a database is down because it runs into a catastrophic problem, millions of dollars in revenue is lost because a company can no longer operate.
- *Eliminate redo apply delay on standby database.* Flashback Database is seamlessly integrated with Data Guard. A standby database can now be quickly and easily flashed back to any point in time so there is no need for a delay in the redo apply.
- *Unanticipated error correction.* Flashback Database provides a continuous snapshot of the Oracle database. The database can be rewound back to an SCN or timestamp.

Requirements

To enable the Flashback Database capability, a DBA configures the Flash Recovery Area. The Flash Recovery Area is a new feature in Oracle Database 10g that provides a common storage location for all recovery-related files and activities in an Oracle database. Besides Flashback Database logs, the recovery area may possibly contain redo archive logs and RMAN backups if the database was configured in that way.

SAP databases take a different approach to storing these kinds of files. For more information on the Flash Recovery Area, consult the Oracle Backup and Recovery Documentation.

- Database has to run in ARCHIVELOG mode
- Set parameters `db_recovery_file_dest_size`, `db_recovery_file_dest`, `db_flashback_retention_target`
- A file system with 30-50 GB space for Flashback Log data

Aim of the document

The document indicates all the steps needed to successfully install and configure Oracle Flashback Database on a physical standby database under SAP conditions. The following SAP notes describe how to set up Flashback Database in an SAP environment and when using BR*Tools.

All terms, like restore points and flashback logging, are explained in these notes and examples of them provided. For Oracle 11g there are no major changes at this points, so the SAP note will be still valid for use.

[966117 Oracle Database 10g: Flashback Database](#)

[966073 Oracle Database 10g: Flash Recovery Area](#)

[937492 FAQ: Oracle Flashback](#)

This document only covers the special scenario of installing Flashback Database on a physical standby database.

Setting up flashback logging

Flashback use is approved in SAP note 105047. For operation the database needs an additional “Flash Recovery Area” memory area which following the conventions in note 966073 is called “oraflash” and included in the Oracle Base directory of the SAP installation.

Given the size needed, the directory should be mounted as an independent file system and included in the directory tree.

Configuration

Firstly define an area of approx. 30-50GB in the file system for storing Flashback Logs. Following the SAP recommendation, the directory

```
/oracle/<sid>/oraflash
```

is created as a “Flash Recovery Area”.

The Flash Recovery Area is also needed on the primary side, but here the “Flashback” feature is not activated. This is necessary because the flashback concept means that the function of flashback logging follows the corresponding standby database.

(see also chapter [Flashback and switchover](#)).

The following parameters should therefore be set on the two databases:

```
DB_RECOVERY_FILE_DEST_SIZE=30G
DB_RECOVERY_FILE_DEST='/oracle/O11/ORAF flash'
DB_FLASHBACK_RETENTION_TARGET=1440;
```

PRIMARY DATABASE	STANDBY DATABASE
SQL> alter system set db_recovery_file_dest_size=30G;	SQL> alter system set db_recovery_file_dest_size=30G;
SQL> alter system set db_recovery_file_dest= '/oracle/O11/oraflash';	SQL> alter system set db_recovery_file_dest= '/oracle/O11/oraflash';
SQL> alter system set db_flashback_retention_target=1440;	SQL> alter system set db_flashback_retention_target=1440;

The area size can be changed dynamically so it can be adjusted to the actual requirements later on. The database continually monitors how full it is and issues a warning in the Alert.log if the area is filling up.

Formulas for estimating a useful Flash Recovery Area size depend on whether:

- Your database has a small or large number of data blocks that change frequently.
- You store backups only on disk, or on disk and tape.
- You use a redundancy-based backup retention policy, or a recovery window-based retention policy.
- You plan to use Flashback Database or a guaranteed restore point as alternatives to point-in-time recovery.

If you plan to enable flashback logging, then note that the volume of Flashback Log generation is approximately the same order of magnitude as redo log generation. For example, if you intend to set `DB_FLASHBACK_RETENTION_TARGET` to 24 hours, and if the database generates 20 GB of redo a day, then as a rule of thumb you should allow 20 GB to 30 GB of disk space for the Flashback Logs. The same rule applies to guaranteed restore points when flashback logging is enabled. For example, if the database generates 20 GB of redo every day, and if the guaranteed restore point will be kept for one day, then plan to allocate 20 to 30 GB.

By way of an example for estimating the disk quota needed, calculate:

Disk Quota =

```
[Size of a copy of database + ]
[Size of an incremental backup + ]
[Size of (n+1) days of archived redo logs + ]
[Size of (y+1) days of foreign archived redo logs (for logical standby) + ]
[Size of control file + ]
[Size of an online redo log member * number of log groups + ]
```

Size of Flashback Logs (based on `DB_FLASHBACK_RETENTION_TARGET` value)

In a typical SAP installation, backups and archive logs have no role to play in the Flash Recovery Area. The size is therefore only affected by `DB_FLASHBACK_RETENTION_TARGET`.

The `V$FLASHBACK_DATABASE_LOG` view offers another way of estimating the size.

```
SQL> SELECT ESTIMATED_FLASHBACK_SIZE FROM V$FLASHBACK_DATABASE_LOG;
```

However, this query only provides usable data once flashback logging has been activated and the database has already run for some time under normal load. Use the data provided for dynamic adjustment of the Flash Recovery Area size.

```
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE=45G;
```

How far back can the database be taken?

The amount of flashback data which can be stored in the flashback area depends on two factors:

- The size of this area, set by `DB_RECOVERY_FILE_DEST_SIZE`
- The retention time, set by `DB_FLASHBACK_RETENTION_TARGET`

If the retention target is reached and the area is not yet full, the database will continue to store Flashback Logs in this area until it reaches the size limit.

If the size limit is reached before the retention time lapses, the database will stop and the DBA will have to resize the area.

If the retention target and also the size limit are reached, the database will start deleting the oldest logs.

The best approach is to find a good size for both parameters which match your business needs.

Activating flashback logging on the standby database

Flashback operations, such as activating or accessing restore points, can only be undertaken in the MOUNT stage of the database. An active recovery will also first have to be canceled.

Manual method

First cancel the recovery on the **standby database**:

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

Then activate Flashback:

```
ALTER DATABASE FLASHBACK ON;
```

Then restart the recovery:

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION
THROUGH LAST SWITCHOVER;
```

With Data Guard Broker

When using Data Guard Broker it is more convenient to deactivate the Broker, restart the database, then activate flashback mode and restart the Broker.

During this process, communication with the primary side is terminated in the correct way and media recovery automatically restarted as soon as the Broker is reactivated.

Broker configuration looks like:

```
DGMGRL> show configuration
```

```
Configuration - o11
```

```
Protection Mode: MaxPerformance
```

```
Databases:
```

```
o11_south - Primary database
```

```
o11_north - Physical standby database
```

```
DGMGRL> edit database o11_south set state = 'LOG-APPLY-OFF';
```

```
Standby DB:
```

```
SQL> CONNECT / AS SYSDBA
```

```
SQL> ALTER DATABASE FLASHBACK ON;
```

```
DGMGRL> edit database o11_south set state = 'ONLINE';
```

Verifying flashback mode

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

```
FLASHBACK_ON
```

```
-----
```

```
YES
```

Other options for this query are:

- NO: Flashback is deactivated
- RESTORE POINT ONLY: Flashback is deactivated but there are guaranteed restore points, so Flashback Logs continue to be stored. Removing the “db_recovery_file_dest” parameter results in an error message the next time the database is started up.

Deactivating

To deactivate flashback mode, use the command:

```
ALTER DATABASE FLASHBACK OFF;
```

All Flashback Logs from the area are subsequently deleted automatically. Flashback is no longer possible until the function has been reactivated and new logs have been created.

Exception: If [guaranteed restore points](#) have been created, the area is not purged.

Flashback logging can be closed in any database mode. There is no need to first shut down and mount, i.e. deactivation is possible at any time even with an open database.

Guaranteed restore points can be identified with the following query:

```
Connect / as sysdba
```

```
SQL> SELECT NAME, SCN, STORAGE_SIZE
       FROM V$RESTORE_POINT
       WHERE GUARANTEE_FLASHBACK_DATABASE = 'YES';
```

Resetting the physical standby database to a defined point in time

This scenario is needed for e.g. partial restoration of a table whose data was changed or even deleted after a certain point in time.

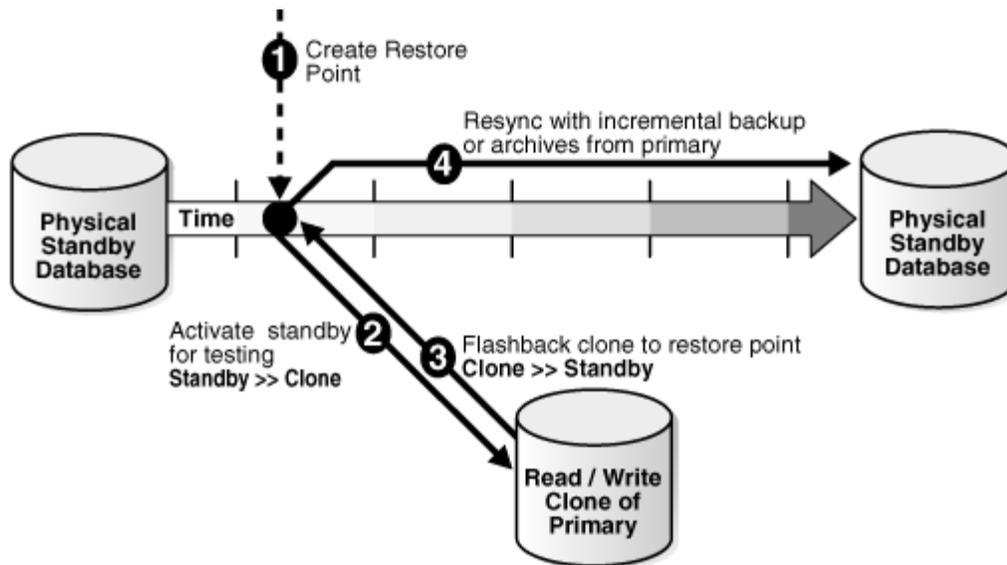
A point-in-time recovery is needed to achieve this with standard methods.

Combining Oracle Data Guard and Flashback offers a convenient and rapid solution.

To find out how far back you can go using Flashback, refer to details of the [oldest flashback SCN](#) in the database:

```
SQL> SELECT OLDEST_FLASHBACK_SCN,
           TO_CHAR(OLDEST_FLASHBACK_TIME, 'DD.MM.YYYY HH24:MI:SS') OLDEST_FLASHBACK_TIME
       FROM V$FLASHBACK_DATABASE_LOG;
```

The following diagram explains the procedure:



In the Oracle documentation this scenario (or similar) is described as “[Using a Physical Standby Database for Read/Write Testing and Reporting](#)”. The following points should be noted:

- A **restore point** must be defined for the subsequent point to go back to.
- The log destination on the primary database must be deactivated to prevent unnecessary error messages caused by the standby database being intermittently down.
- The standby database is opened.
- Data can be extracted using EXP or SQLPLUS.
- The open database is changed back into a standby database.
- All archives produced up to this point are reconstructed.
- The log destination on the primary database is reactivated.
- The standby database restarts regular operation.

This process intermittently places the standby database in the past without it however losing its ability to run as a standby database again later on.

However, during this period the primary database has no standby protection unless another physical standby database is set up and the aforementioned processes are undertaken on it.

Example:

The ZMAT table on the production database was partially deleted because of a failed transport job and it has been partly updated with incorrect data. The ZMATBEZ table has been dropped.

You can narrow the time of the incident down to 01.10.2008 at around 11:10 am. The redo has already been applied to the standby database.

Flashback logging was activated on the standby database and you now want to extract the lost data from the standby database.

Use the log switches to determine the log sequence and SCN affected. Flashback can take you back to either an SCN or timestamp. Under certain circumstances, the timestamp you are looking for will not be found in the Flashback Logs so going back to an SCN is usually the safer approach.

PRIMARY DATABASE	STANDBY DATABASE
11:08 Logswitch SCN 10585218, Seq 16182	
11:10 Transport task: DELETE TABLE ZMAT... WHERE... UPDATE ZMAT SET ... DROP TABLE ZMATBEZ;	
11:13 Logswitch Seq: 16183	
	16:08 Redo Apply SCN 10585218, Seq: 16182
	16:13 Redo Apply Seq: 16183
17:45 The incident is recognized	

Defining a guaranteed restore point

Stop the recovery, otherwise you cannot set a restore point and this results in the error message:

```
ORA-38784: Cannot create restore point...
ORA-01153: an incompatible media recovery is active
```

Then define a restore point for later resetting the standby database to the current position.

PRIMARY DATABASE	STANDBY DATABASE
	<pre>DGMGRL> EDIT DATABASE o11_south set state ='LOG-APPLY-OFF'; SQL> CREATE RESTORE POINT before_flashback GUARANTEE FLASHBACK DATABASE;</pre>

Deactivating log transport

By deactivating the LOG_ARCHIVE_DEST_2 parameter, Broker error messages are avoided because the standby database no longer responds.

Alternatively the Broker can also be stopped if no other standby database is configured.

PRIMARY DATABASE	STANDBY DATABASE
<pre>SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;</pre>	<pre>DGMGRL> EDIT DATABASE o11_north SET STATE=' LOG-TRANSPORT-OFF' ;</pre>

Opening the standby database

Once the standby database has been activated, it needs to be restarted. The protection mode should be set to MAXIMUM PERFORMANCE to rule out the primary database being impaired if MAXIMUM AVAILABILITY or MAXIMUM PROTECTION has previously been activated.

PRIMARY DATABASE	STANDBY DATABASE
	<pre>ALTER DATABASE ACTIVATE STANDBY DATABASE;</pre>
	<pre>SHUTDOWN IMMEDIATE;</pre>
	<pre>STARTUP MOUNT;</pre>
	<pre>ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE;</pre>
	<pre>ALTER DATABASE OPEN; SELECT DATABASE_ROLE FROM V\$DATABASE;</pre>
	<pre>DATABASE_ROLE ----- PRIMARY</pre>

Resetting the standby database to the desired point in time

The appropriate point in time is as soon as possible before the failure occurred on the primary database. An accurate SCN is not normally known so it is a good idea to select the SCN of the last log switch before the event or the timestamp before the start of the identified transport job. The tables would still have been in their original condition at this time.

Syntax:

- SCN
FLASHBACK [STANDBY] DATABASE [<database_name>] TO [BEFORE] SCN
<system_change_number>
- TIMESTAMP
FLASHBACK [STANDBY] DATABASE [<database_name>] TO [BEFORE] TIMESTMP
<system_timestamp_value>

PRIMARY DATABASE	STANDBY DATABASE
	SHUTDOWN IMMEDIATE;
	STARTUP MOUNT; FLASHBACK DATABASE TO SCN 10585218; Possible as an alternative: FLASHBACK DATABASE TO TIMESTAMP timestamp '2008-10-01 11:08:18';
	ALTER DATABASE OPEN READ ONLY;

The steps required to extract the data can now be carried out. You should check whether the data contains the desired information. If not the flashback must be reverted to an earlier point in time (SCN).

Extracting the data

Use e.g. the BR*Tools for a table export or run it manually. Please note that the specialist department will always have to run a consistency check.

PRIMARY DATABASE	STANDBY DATABASE
	host> exp file=ZMAT rows=y tables=ZMAT
	host> exp file=ZMATBEZ rows=y tables=ZMATBEZ
<p>Read in data by importing into a separate user schema and compare with the existing table.</p> <p>In the event of a dropped table, it can be re-established in the original user schema by importing.</p>	

Returning the open database to a standby role

The database is returned to the last SCN before the recovery was stopped by going to the previously set guaranteed restore point.

Syntax:

- ```
RESTORE POINT
FLASHBACK [STANDBY] DATABASE [<database_name>] TO [BEFORE] RESTORE
POINT <restore_point_name>
```

| PRIMARY DATABASE | STANDBY DATABASE                                                                                                                                                                                                                    |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | <pre>STARTUP MOUNT FORCE; FLASHBACK DATABASE TO RESTORE POINT before_flashback; ALTER DATABASE CONVERT TO PHYSICAL STANDBY; STARTUP MOUNT FORCE;  SELECT DATABASE_ROLE FROM V\$DATABASE  DATABASE_ROLE ----- PHYSICAL STANDBY</pre> |

Reactivating Logapply on the standby database

| PRIMARY DATABASE                                                  | STANDBY DATABASE                                                                                                                                                                                                                   |
|-------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>DGMGRL&gt; EDIT DATABASE o11_north SET STATE='ONLINE';</pre> | <pre>DGMGRL&gt; EDIT DATABASE o11_south SET STATE='ONLINE'</pre> <p>Alternatively, if DG Broker is not being used:</p> <pre>ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION THROUGH LAST SWITCHOVER;</pre> |

Activating the Broker state reactivates the temporarily deactivated log\_archive\_dest\_2 on the primary system and starts recovery on the standby database.

If necessary any “gaps” in the log sequence are resolved using FAL requests to the primary host and the archives are automatically reconstructed:

```
Alert.log:

Fetching gap sequence in thread 1, gap sequence 471-489

Redo Shipping Client Connected as PUBLIC
-- Connected User is Valid
RFS[2]: Assigned to RFS process 19420
RFS[2]: Identified database type as 'physical standby'
RFS[2]: Archived Log: '/oracle/DGS/oraarch/1_475_652890833.dbf'
RFS[2]: Archived Log: '/oracle/DGS/oraarch/1_476_652890833.dbf'
RFS[2]: Archived Log: '/oracle/DGS/oraarch/1_477_652890833.dbf'

...
Media Recovery Log /oracle/DGS/oraarch/1_475_652890833.dbf
Media Recovery Log /oracle/DGS/oraarch/1_476_652890833.dbf
Media Recovery Log /oracle/DGS/oraarch/1_477_652890833.dbf

...
Media Recovery Log /oracle/DGS/oraarch/1_489_652890833.dbf
```

If the gaps cannot be automatically resolved because under certain circumstances the archives have already been backed up by BR\*Archive, these have to be manually copied back to the log\_archive\_dest\_1 on the primary side. In most cases a FAL request can then be successfully carried out. **Oracle® Data Guard Concepts and Administration** chapter [12.11.2 Determining If an Archive Gap Exists](#) contains useful information on how to manually rectify archive gaps.

## Deleting the guaranteed restore point

---

PRIMARY DATABASE

STANDBY DATABASE

---

DROP RESTORE POINT before\_flashback;

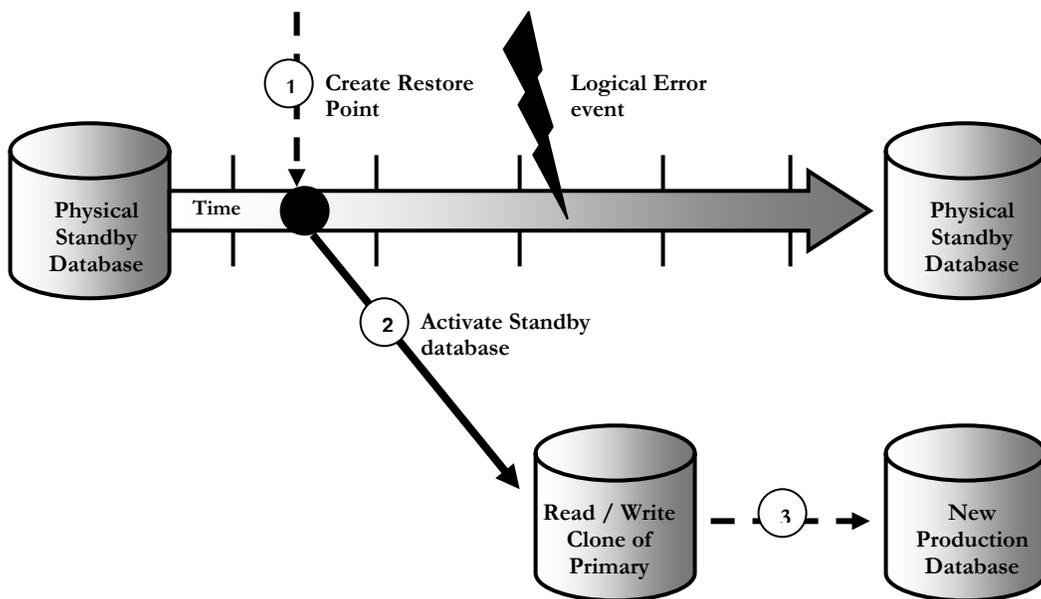
---

## Using physical standby as primary database after flashback

This scenario is needed if the primary host fails, i.e. if a FAILOVER occurs and the standby database cannot be used in its current state because of another “human error”. This situation is similar to a point-in-time recovery.

The procedure is similar to that described under [Resetting the physical standby database to a defined point in time](#).

Only the step of resetting the open standby database is omitted and it is opened in READ WRITE mode.



The following points should be noted:

- Set restore point on standby database
- Open the standby database.
- Define timestamp / SCN for flashback
- Reset database

- Use database as new production database  
After a logical error on the production database, the standby database is reset and used as the new production database (before the logical error).

## Setting the restore point

```
DGMGRL> EDIT DATABASE o11_south SET STATE='LOG-APPLY-OFF';
DGMGRL> EDIT DATABASE o11_north SET STATE='LOG-TRANSPORT-OFF';

SQL> CREATE RESTORE POINT before_flashback GUARANTEE FLASHBACK DATABASE;
```

## Opening the standby database

```
SQL> ALTER DATABASE ACTIVATE STANDBY DATABASE;
SQL> STARTUP MOUNT FORCE;
SQL> ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE;
SQL> ALTER DATABASE OPEN;
```

## Finding the right time for the flashback and resetting the database

The potential point in time selected depends on the error situation. Possible SCNs for going back can be taken from the alert log of the primary database.

```
SQL> SHUTDOWN IMMEDIATE;

SQL> STARTUP MOUNT;

SQL> FLASHBACK DATABASE TO SCN 46835871;
```

Or alternatively:

```
SQL> FLASHBACK DATABASE TO TIMESTAMP timestamp '2008-12-14 06:22:46';

SQL> ALTER DATABASE OPEN READ WRITE;
```

## Using the database as the new production database

```
SQL> SHUTDOWN IMMEDIATE;
```

```
SQL> STARTUP;
```

The “old” standby database can now be used as the “new” production database.

## Flashback and switchover

If Flashback Database is to remain active for the (current) standby database after a switchover, a Flash Recovery Area should also be created here.

```
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE=30G;
```

```
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='/oracle/O11/oraflash';
```

The steps described in chapter [Activating Flashback Database on standby database](#) should still be carried out.

Flashback must be deactivated on what is now the current primary database. Otherwise Flashback Logs will continue to be created and this would result in marked drops in performance on the production database. Therefore deactivate Flashback manually after the switchover.

```
SQL> ALTER DATABASE FLASHBACK OFF;
```

This state must be reversed after another switchover back to the old configuration and flashback only activated for the standby database.

## Flashback and failover

After a failover, the database previously used as the standby database becomes the primary database. Activation of flashback logging is retained after a failover. In this case deactivate Flashback to avoid further drops in performance. (Also refer to Deactivating flashback.)

```
SQL> ALTER DATABASE FLASHBACK OFF;
```

## Annex

### Database parameters

```
DB_RECOVERY_FILE_DEST_SIZE=30G
DB_RECOVERY_FILE_DEST='/oracle/O11/ORAFFLASH'
DB_FLASHBACK_RETENTION_TARGET=1440;
```

### Guaranteed restore points

```
SELECT NAME, SCN, STORAGE_SIZE
FROM V$RESTORE_POINT
WHERE GUARANTEE_FLASHBACK_DATABASE = 'YES';
```

### Oldest possible SCN for restore point

```
SELECT OLDEST_FLASHBACK_SCN,
TO_CHAR(OLDEST_FLASHBACK_TIME, 'DD.MM.YYYY HH24:MI:SS')
OLDEST_FLASHBACK_TIME
FROM V$FLASHBACK_DATABASE_LOG;
```

### Creating the standard restore point

```
CREATE RESTORE POINT <RESTORE POINT_NAME>;
```

### Creating the guaranteed restore point

```
CREATE RESTORE POINT <RESTORE POINT NAME> GUARANTEE FLASHBACK DATABASE;
```

### Memory used for all guaranteed restore points

```
SELECT SUM(STORAGE_SIZE)
FROM V$RESTORE_POINT
WHERE GUARANTEE_FLASHBACK_DATABASE = 'YES';
```

## Memory used per guaranteed restore point

```
SELECT NAME, SCN, STORAGE_SIZE
FROM V$RESTORE_POINT
WHERE GUARANTEE_FLASHBACK_DATABASE = 'YES';
```

## Deleting the restore point

```
DROP RESTORE POINT <RESTORE POINT NAME>;
```



Setup Flashback Database on Data Guard  
physical standby database for SAP customers  
January 2010

Author: Bernhard Baumann  
Oracle Support Solution Center SAP,  
Walldorf (Germany)

Contributing Authors: Tammy Bednar in 2003

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

Copyright © 2009, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.