webMIP database - backup, recovery and failover procedures

Andrew Hardy

December 18, 2008

Abstract

This document outlines the procedures used with the webMIP database with regards to backup and recovery

Contents

Listings

Chapter 1

Overview

1.1 Requirements

The primary requirements for the backup and recovery of the webMIP system are:

1. Support the customer Service Level Agreement(SLA) of having data-loss at less than 4 hours within the working-day (8am to 5pm);
2. Domain Name Server(DNS);

1.2 System Architecture

1.2.1 Primary and standby databases

1.2.2 Internet access

Chapter 2

Backup

2.1 COLD backup

2.2 HOT backup

A Hot backup is performed against an actively running database i.e. one in which users are connected and transactions are occurring. To use Hot backups you must operate the database in ARCHIVELOG mode (see listing 2.2). Recovery from a Hot backup relies on the restoration of the datafiles and the replaying of all archive logs from the point at which the original backup and onwards.

Listing 2.1: Invoking SQL*Plus

```
1 REM Identify the database connection as
2 REM bequeath by using the database SID
3 REM Invoke SQL*Plus without logging in
4 set oracle_sid=WEBMIP
5 sqlplus /nolog
```

Listing 2.2: Prepare for archivelog

```
1 REM $Id: archivelog.sql 6472 2008-08-28 09:56:01Z hardya $
2 conn &&user/&&password as sysdba;
3 shutdown immediate;
4 conn &&user/&&password as sysdba;
5 startup mount;
6 alter database force logging;
7 alter system set log_archive_dest_1='location=s:\orabackup\
    webmip\archivelogs\' scope=both;
8 /*
9 set the time by which archive logs
10 should lag the redo in seconds
11 */
12 alter system set archive_lag_target=2700;
13 alter database archivelog;
14 alter database open;
```

Once in ARCHIVELOG mode, the datafiles and control files are made available to be backed-up at the operating system level. The generated archive logs from the point at which the backup started are also backed up. In listing 2.3, the first 'archive log list;' is used to provide details of the oldest online log sequence number at the start of the backup whilst the second 'archive log list;' is used to provide the *current* log sequence number at the end of the backup. To recover the database to a state consistent with the end of the backup, you *must* include all the archived log files starting from oldest sequence number to the current sequence number as part of the backup [1]. The 'alter system switch logfile;' command forces a log switch so that the *current* log is archived[2] and so made available to be backed up. You should *not* backup and restore the online redo logs as this will prevent the application of the archive logs during recovery. You *cannot* backup and restore files associated with temporary tablespaces. Instead, these files are recreated using the 'create_tempfiles.sql' script generated by the hot backup script[3].

Listing 2.3: Perform HOT backup

```
1  REM $Id: hotbackup.sql 6472 2008-08-28 09:56:01Z hardya $
2  SET SERVEROUTPUT ON
3  store set set_settings.sql
4  set feedback off pagesize 0 heading off verify off
5  set linesize 100 trimspool on
6  DEFINE backup_dir = 's:\orabackup\webmip\files\'
7  REM Determine the oldest archive log to be backed up
8  archive log list;
9  alter system switch logfile;
10 REM
11 REM produce a SQL script that:
12 REM   places each tablespace into 'backup' mode,
13 REM   copies the contents of the tablespace to the
14 REM     backup directory,
15 REM   returns each tablespace from 'backup' mode
16 REM
17 PROMPT *** SPOOLING
18 spool do_backup.sql
19 WITH tsp_df AS (
20 SELECT tsp.tablespace_name
21        ,df.file_name
22        ,rownum AS current_row
23        ,first_value(rownum) over(PARTITION BY tsp.
               tablespace_name) AS first_row
24        ,last_value(rownum) over(PARTITION BY tsp.
               tablespace_name) AS last_row
25   FROM dba_tablespaces tsp
26        ,dba_data_files  df
27  WHERE tsp.contents <> 'TEMPORARY'
28    AND tsp.tablespace_name = df.tablespace_name
29 )
```

[1] Recovery will start from the point of the 'current' log, however as a precautionary measure you should keep all archived log files starting from the *oldest online log sequence* number.

[2] The creation of archive logs lags the online redo logs.

[3] Need to confirm this

```
30 SELECT cmd
31   FROM (SELECT 'alter␣tablespace␣' || tablespace_name || '␣
         begin␣backup;' AS cmd
32                ,tablespace_name
33                ,1 AS seq
34           FROM tsp_df
35          WHERE current_row = first_row
36         UNION
37         SELECT 'host␣ocopy␣' || file_name || '␣&&backup_dir'
               AS cmd
38                ,tablespace_name
39                ,2 AS seq
40           FROM tsp_df
41          WHERE (current_row = first_row)
42             OR (current_row <> first_row AND current_row <>
                   last_row)
43             OR (current_row = last_row AND last_row <>
                   first_row)
44         UNION
45         SELECT 'alter␣tablespace␣' || tablespace_name || '␣end
               ␣backup;' AS cmd
46                ,tablespace_name
47                ,3 AS seq
48           FROM tsp_df
49          WHERE current_row = last_row)
50  ORDER BY tablespace_name
51           ,seq
52 /
53 spool off;
54 PROMPT *** SPOOLING COMPLETE
55 @set_settings
56 @@do_backup
57 REM
58 REM Create SQL script to add temporary files to temporary
       tablespaces
59 REM
60 set feedback off pagesize 0 heading off verify off
61 set linesize 100 trimspool on
62 spool create_tempfile$.sql
63 select 'alter␣tablespace␣'||ts.name||'␣add␣tempfile␣'''||df.
       name||'''␣size␣'|| df.bytes||'␣reuse;' from v$tempfile df,
        v$tablespace ts
64 where ts.ts# = df.ts#
65 /
66 spool off
67 host move create_tempfile$.sql &&backupdir\create_tempfiles.
       sql
68 @set_settings
69 REM
70 REM create control file for the standby database
71 REM create backup control file for the primary database
72 REM
73 alter database create standby controlfile as 's:\orabackup\
       webmip\files\standby.ctl';
```

```
74 alter database backup controlfile to 's:\orabackup\webmip\
       files\backup.ctl';
75 REM
76 REM create a database initialization script for
77 REM the standby database
78 REM
79 create pfile='s:\orabackup\webmip\files\initstandby.ora' from
       spfile;
80 REM
81 REM Determine the current archive log to be backed up
82 archive log list;
83 alter system switch logfile;
```

You also need a copy of the password file that allows SYSDBA access[4], this should be located in the 'database' directory of the Oracle installation e.g. 'C:\oracle\product\11.1.0\db_1\database' as 'PWDwebmip.ora'.

Listing 2.4: Copying the password file to the backup location

```
1 copy c:\oracle\product\11.1.0\db\_1\database\PWDwebmip.ora s:\
      orabackup\webmip\files
```

At this point you have a set of datafiles, archive logs and control files that can be used to:

1. Recover the primary database (see section 3.2);
2. Create a standby database (see section 4.1.2).

These files should be stored outside of the server environment.

[4]The password file should be re-copied whenever the passwords on the primary database are altered

Chapter 3

Recovery

3.1 COLD recovery

3.2 HOT recovery

Chapter 4

Standby database

A standby database is a maintained duplicate, or 'standby', of the production or 'primary' database for recovering from disasters at the primary site. The intention is to be able to switch over from the primary database to the standby database in the case of disaster in the least amount of time and with as little recovery as possible. A hot standby database is a backup copy of the primary database that is maintained on a separate machine. A Hot backup or Cold backup is made of the primary database and copied to the standby server. The standby database is mounted but not opened. The archive logs from the primary database are copied from the primary to the standby database and applied at regular intervals. The means that the standby database is always a few log files (at least one log file) behind the primary database and is always in 'mounted but not open' stage.

When the primary database fails, the standby database can be opened and all users switched to the standby server. After such a switch, the standby database becomes the primary database. A new standby database will then be required.

4.1 Creation

4.1.1 Preparation on the primary database

The primary database is prepared by being placed into ARCHIVELOG mode and a Hot backup being performed (see section 2.2 for details on this). The 'alter database create standby control file...' command in listing 2.3 produces the control file that will be used by the standby database, whilst the 'create pfile...' command produces a text version of the database initialization file.

4.1.2 Creation of the standby database

The standby database is created from a Hot backup of the primary database (see section 2.2). The datafiles are copied to the standby server. The standby server should have the same directory structure as the primary server in order to minimize the amount of changes to be made on the standby server (check the contents of the 'initstandby.ora' created in listing 2.3 for the location of the 'db_recovery_file_dest' as this is generally only created when databases are

created using the Oracle Database Configuration Agent). Do not forget to copy the 'PWDwebmip.ora' password file to the correct 'database' location (listing 4.1).

Listing 4.1: Copying the password file to the standby location

```
1 copy s:\orabackup\webmip\files\PWDwebmip.ora c:\oracle\product
    \11.1.0\db\_1\database\PWDwebmip.ora
```

The 'oradim' command (listing 4.2) is used to create the Windows service instance to support the standby database. The service is created with a 'manual' start mode - this means that the database instance will only become available when started manually.

Listing 4.2: Creation of Windows service for standby database

```
1 oradim -NEW -SID webmip -STARTMODE manual
```

The 'Oracle Net Manager' tool is used to create a 'listener' for the database and the listener restarted to reflect the changes (see listing 4.3).

Listing 4.3: Restarting the standby listener

```
1 lsnrctl stop
2 lsnrctl start
```

Where there are differences between the configuration of the primary and standby servers, the 'initstandby.ora' database initialization file generated by listing 2.3 is altered. The 'standby.ctl' file (also created by 2.3) is copied to the appropriate directory. It is recommended to make multiple copies of this control file for greater resilience. Ensure that the names used for the control files are consistent with those listed in the 'initstandby.ora'.

The standby database is now physically present, but the instance is not running. The 'oradim' command (see listing 4.4) is used to start the database service, but *not* the instance. The 'ORACLE_SID' environment variable is used to identify the database instance that we are about to start - again we connect to SQL*Plus with the 'nolog' option due to the difficulties in creating the connection string from the Windows command line.

Listing 4.4: Starting the standby service

```
1 REM $Id: oradim_start.cmd 6472 2008-08-28 09:56:01Z hardya $
2 oradim -STARTUP -SID webmip -STARTTYPE srvc
```

We can now connect to the 'idle' instance and start it ready to receive archive logs from the primary server (listing 4.5[1]). We create the database 'spfile' from the modified 'initstandby.ora' file and startup the database as a physical standby, but without recovering[2].

[1] If you get a permissions error, connect with the sys password i.e. conn sys/'webmip' as sysdba

[2] Hot backups always require recovery using the archive logs that were generated during the period of the backup.

Listing 4.5: Starting the standby database

```
conn / as sysdba
create spfile from pfile='s:\orabackup\webmip\files\
    initstandby.ora';
startup nomount;
alter database mount standby database;
```

At this point the standby database is ready to be 'maintained' (see section 4.2).

4.2 Maintaining standby database

The standby database is physically present with the datafiles as copied from the primary database. The database has been started up, but recovery has not taken place. Recovery requires the applying of archive logs to the database: if the archive logs generated during the original backup were applied, then the recovered database would be consistent with the state of the primary database as it was at the end of the backup period. A standby database extends this recovery mechanism by applying not only the archive logs generate during the backup, but some or all of the archive logs generate by the primary database since the time of the backup. This is achieved by keeping the standby database unmounted and periodically attempting recovery using archive logs delivered from the primary database.

4.2.1 Transfer of archive logs from primary database

A mechanism is required to periodically transfer archive logs generated by the primary database to the archive log destination of the standby server. The VB-Script 'logship.vbs' (listing 4.6) does this by comparing the archive logs available on the primary server with those available on the standby and transferring any missing files.

Listing 4.6: Transferring archive logs

```
REM $Id: advwb1_logship.vbs 6472 2008-08-28 09:56:01Z hardya $
const archive_source = "\\advweb\archivelogs"
const logship_user = "advweb\logship"
const logship_pwd = "logship"
const archive_dest = "\\advwb2.opcs-online.info\archivelogs"
const strFrom = "webmip@AdvanticaGroup.com"
const strTo = "Andrew.Hardy@AdvanticaGroup.com"
const strSub = "MIP: Logship"
const strSubError = "MIP: Logship Error"
strBody = "Logship Completed Successfully"
const strSMTP = "relay01.sleek.net"


CopyArchives archive_source,archive_dest, logship_user,
    logship_pwd

Sub CopyArchives(strSource, strDest, strUser, strPwd)
        Dim strLocalDrive
```

```
18          strLocalDrive = "L:"
19
20          Dim strRemoteDrive
21          strRemoteDrive = "R:"
22
23          'error handling
24          On Error Resume Next
25
26          Set objNetwork = CreateObject("WScript.Network")
27
28          objNetwork.MapNetworkDrive strLocalDrive, strSource,
                false, strUser, strPwd
29
30          'handle error
31          If Err.Number <> 0 Then
32                  SendEmail strFrom, StrTo, strSubError, "Error
                        Mapping " & strSource & ":" & Err.
                        Description, StrSMTP
33                  Wscript.Echo "Error number " & Err.Number & "
                        occured: " & Err.Description
34                  Wscript.Echo ""
35                  Wscript.Quit Err.Number
36          End If
37
38          objNetwork.MapNetworkDrive strRemoteDrive, strDest,
                false, strUser, strPwd
39
40
41          'handle error
42          If Err.Number <> 0 Then
43                  SendEmail strFrom, StrTo, strSubError, "Error
                        Mapping " & strDest & ":" & Err.
                        Description, StrSMTP
44                  Wscript.Echo "Error number " & Err.Number & "
                        occured: " & Err.Description
45                  Wscript.Echo ""
46                  Wscript.Quit Err.Number
47          End If
48
49          set fileSystem = createobject("Scripting.
                FileSystemObject")
50
51          WScript.Echo "Getting fldrSource"
52          set fldrSource = fileSystem.GetFolder(strLocalDrive)
53
54          'handle error
55          If Err.Number <> 0 Then
56                  SendEmail strFrom, StrTo, strSubError, "Error
                        Getting Local Folder " & strLocalDrive & "
                        :" & Err.Description, StrSMTP
57                  Wscript.Echo "Error number " & Err.Number & "
                        occured: " & Err.Description
58                  Wscript.Echo ""
59                  Wscript.Quit Err.Number
```

```
60          End If
61
62
63          WScript.Echo "Getting fldrDest"
64          set fldrDest = fileSystem.GetFolder(strRemoteDrive)
65
66          'handle error
67          If Err.Number <> 0 Then
68                  SendEmail strFrom, StrTo, strSubError, "Error
                        Getting Remote Folder " & strRemoteDrive &
                         ":" & Err.Description, StrSMTP
69                  Wscript.Echo "Error number " & Err.Number & "
                        occured: " & Err.Description
70                  Wscript.Echo ""
71                  Wscript.Quit Err.Number
72          End If
73
74
75          WScript.Echo "Looking at directory " & strSource
76          for each file in fldrSource.Files
77                  if not filesystem.FileExists(strDest & "\" &
                        file.Name) then
78                          WScript.Echo "Copying " & file.Name &
                                " to " & strDest & "\" & file.Name
79                          file.Copy (strDest & "\" & file.Name)
80                  end if
81          next
82
83          'handle error
84          If Err.Number <> 0 Then
85                  SendEmail strFrom, StrTo, strSubError, "Error
                        During File Copy:" & Err.Description,
                        StrSMTP
86                  Wscript.Echo "Error number " & Err.Number & "
                        occured: " & Err.Description
87                  Wscript.Echo ""
88                  Wscript.Quit Err.Number
89          End If
90
91          objNetwork.RemoveNetworkDrive strRemoteDrive
92          'handle error
93          If Err.Number <> 0 Then
94                  SendEmail strFrom, StrTo, strSubError, "Error
                        Removing Remote Folder " & strRemoteDrive
                        & ":" & Err.Description, StrSMTP
95                  Wscript.Echo "Error number " & Err.Number & "
                        occured: " & Err.Description
96                  Wscript.Echo ""
97                  Wscript.Quit Err.Number
98          End If
99
100         objNetwork.RemoveNetworkDrive strLocalDrive
101         'handle error
102         If Err.Number <> 0 Then
```

```
103                     SendEmail strFrom, StrTo, strSubError, "Error
                            Removing Local Folder " & strLocalDrive &
                            ":" & Err.Description, StrSMTP
104                     Wscript.Echo "Error number " & Err.Number & "
                            occured: " & Err.Description
105                     Wscript.Echo ""
106                     Wscript.Quit Err.Number
107             End If
108
109             SendEmail strFrom, StrTo, strSub, strBody, StrSMTP
110
111 End Sub
112
113 ' From the book "Windows Server Cookbook" by Robbie Allen
114 ' ISBN: 0-596-00633-0
115
116 Sub SendEmail(strFrom, StrTo, strSub, strBody, StrSMTP)
117             set objEmail = CreateObject("CDO.Message")
118             objEmail.From = strFrom
119             objEmail.To = strTo
120             objEmail.Subject = strSub
121             objEmail.Textbody = strBody
122             objEmail.Configuration.Fields.Item("http://schemas.
                    microsoft.com/cdo/configuration/sendusing") = 2
123             objEmail.Configuration.Fields.Item("http://schemas.
                    microsoft.com/cdo/configuration/smtpserver") =
                    strSMTP
124             objEmail.Configuration.Fields.Update
125             objEmail.Send
126             If Err.Number <> 0 Then
127                     Wscript.Echo "Error number " & Err.Number & "
                            occured: " & Err.Description
128                     Wscript.Echo ""
129                     Wscript.Quit Err.Number
130             End If
131             WScript.Echo "Email sent"
132 End Sub
```

Windows 'Scheduled Tasks' are used to invoked this VBScript on a regular basis. The period between runs of this script largely determines the lag between the primary and standby databases in the case of a disaster.

4.2.2 Application of archive logs on standby database

The standby database is mounted in a standby mode, users cannot access the database to perform queries, etc. Archive logs from the primary server are applied to the standby database through the standard 'recovery' mechanism - simulating the recovery of a Hot backup, but *without* ending the recovery cycle. Listing 4.7 shows the method of placing the database into recovery mode. In this mode, the database automatically applies the archive logs shipped from the primary server. If the database determines that it requires a missing archive log (one that has not been shipped), it will raise an error. If the database is able to apply all required archive logs, it will continue until the following line where

we 'cancel' the recovery - this allows recovery to continue at a later date.

Listing 4.7: Applying archive logs

```
REM $Id: applylog.sql 6472 2008-08-28 09:56:01Z hardya $
set echo on
spool s:\orabackup\webmip\scripts\applylog.log
connect sys/fmdidgad as sysdba
alter database recover automatic standby database until cancel
    ;
alter database recover cancel;
spool off;
exit
```

Windows 'Scheduled Tasks' are used to invoke the application of the archived logs on a regular basis through the use of a Windows command file (see listing 4.8).

Listing 4.8: Applying archive logs - command file

```
REM $Id: applylog.cmd 6472 2008-08-28 09:56:01Z hardya $
set ORACLE_SID=webmip
sqlplus /nolog @applylog
```

The 'currency' of the application of the archive logs can be checked by reviewing the database view v$log_history on both the primary and standby databases (see listing 4.9).

Listing 4.9: Checking the applying of the archive logs

```
alter session set nls_date_format='DD-MON-YYYY_HH24:MI:SS'
/
SELECT FIRST_TIME, FIRST_CHANGE#, NEXT_CHANGE#, SEQUENCE# FROM
    V$LOG_HISTORY
/
```

4.2.3 Restart of standby server

Whenever the standby server is restarted, the following scripts should be followed:

Listing 4.10: Restarting the standby service

```
REM $Id: oradim_start.cmd 6472 2008-08-28 09:56:01Z hardya $
oradim -STARTUP -SID webmip -STARTTYPE srvc
```

Listing 4.11: Restarting the standby database

```
conn / as sysdba
startup nomount;
alter database mount standby database;
```

At this point the standby database is ready to be 'maintained'.

Chapter 5

Failover

5.1 Activation of standby database

5.1.1 Preparation of the primary database

5.1.2 Preparation of the standby database

5.1.3 Activating the standby database

The SQL commands in listing 5.1 firstly activate the database, then shut it down cleanly prior to restarting and opening the database.

Listing 5.1: Activating standby database

```
1 REM $Id: activatestandby.sql 6472 2008-08-28 09:56:01Z hardya
    $
2 set echo on
3 spool activatestandby.log
4 connect &&user/&&password as sysdba
5 alter database activate standby;
6 shutdown immediate;
7 startup mount;
8 alter database open;
9 spool off;
10 exit
```

On completion, the database is open for read write access and is a duplicate of the primary database to the point at which the last archive log from the primary database was applied. The standby database will generate a new set of archive logs.

5.2 Redirection of DNS

Chapter 6

Failback

6.1 HOT failback

6.1.1 Preparation of the primary database

6.1.2 Backup of the standby database

6.1.3 Transfer of files from standby to primary databases

6.1.4 Maintaining primary database as standby

6.1.5 Re-activation of primary database

6.1.6 Redirection of DNS

6.1.7 Recreation of standby database

6.2 COLD failback

6.2.1 Preparation of the primary database

6.2.2 Backup of the standby database

6.2.3 Transfer of files from standby to primary databases

6.2.4 Re-activation of primary database

6.2.5 Redirection of DNS

6.2.6 Recreation of standby database

Glossary

ARCHIVELOG As Oracle rotates through its Redo log groups, it will eventually overwrite a group which it has already written to. The data that is being overwritten would be useless for a recovery scenario. In order to prevent that, a database can be run in archive log mode. If the database is in log archive mode, the database makes sure that online Redo logs are not overwritten before they have been archived.. 6, 7, 11

Cold backup A cold backup, also called an offline backup, is a database backup when the database is offline and thus not accessible for updating.. 11

Domain Name Server or Service An Internet service that translates domain names into IP addresses.. 5

Hot backup A hot backup, also called an online backup, is a backup performed on data even though it is actively accessible to users and may currently be in a state of being updated.. 6, 11, 12, 16

primary database The primary database is the database accessed by users under normal conditions.. 20

Redo log Before Oracle changes data in a datafile it writes these changes to the redo log. If something happens to one of the datafiles, a backed up datafile can be restored and the redo that was written since that backup is applied bringing the datafile to the state it had before it became unavailable. The same technique is used in a standby databases environment: One database (the primary database) records all changes and sends them to the standby database(s). These standby databases in turn apply the arrived redo and this keeps them synchronized with the primary database.. 20

Service Level Agreement A service level agreement (frequently SLA) is that part of a service contract where the level of service is formally defined. In practice, the term SLA is sometimes used incorrectly in the context of contracted delivery time (of the service) or performance.. 5

standby database A standby database is a maintained duplicate of the primary database.. 11