COMPAQ

White Paper

December 1999
11EE-1199A-WWEN
Prepared by:
Storage Division
Compaq Computer Corporation

Contents

Enterprise Volume Manager and Oracle8 Best Practices

Abstract: The purpose of this document is to define the scope of the integration between Oracle and Compaq SANworks™ Enterprise Volume Manager (EVM). Procedures for using EVM with Oracle, including sample SQL scripts, will be provided. Sufficiently detailed descriptions of both the Oracle architecture and EVM will be provided.

This document is intended for readers that are familiar with Oracle database administration. Many of the commands and functions described within require DBA privileges and can be disruptive to database operation if used incorrectly. If you are not completely comfortable using the commands and functions described herein, consult your Oracle documentation before proceeding.

Notice

The information in this publication is subject to change without notice and is provided "AS IS" WITHOUT WARRANTY OF ANY KIND. THE ENTIRE RISK ARISING OUT OF THE USE OF THIS INFORMATION REMAINS WITH RECIPIENT. IN NO EVENT SHALL COMPAQ BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL, PUNITIVE OR OTHER DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION OR LOSS OF BUSINESS INFORMATION), EVEN IF COMPAQ HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The limited warranties for Compaq products are exclusively set forth in the documentation accompanying such products. Nothing herein should be construed as constituting a further or additional warranty.

This publication does not constitute an endorsement of the product or products that were tested. The configuration or configurations tested or described may or may not be the only available solution. This test is not a determination of product quality or correctness, nor does it ensure compliance with any federal, state or local requirements.

Compaq, Deskpro, Compaq Insight Manager, Systempro, Systempro/LT, ProLiant, ROMPaq, QVision, SmartStart, NetFlex, QuickFind, PaqFax, and Prosignia are registered with the United States Patent and Trademark Office.

ActiveAnswers, Netelligent, Systempro/XL, SoftPaq, Fastart, QuickBlank, QuickLock are trademarks and/or service marks of Compaq Computer Corporation.

Microsoft, Windows and Windows NT are trademarks and/or registered trademarks of Microsoft Corporation.

Intel, Pentium and Xeon are trademarks and/or registered trademarks of Intel Corporation.

Other product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Enterprise Volume Manager and Oracle8 Best Practices
White Paper prepared by Storage Division

First Edition (December 1999)
Document Number 11EE-1199A-WWEN

Using EVM to Back Up an Oracle 8.0 Database

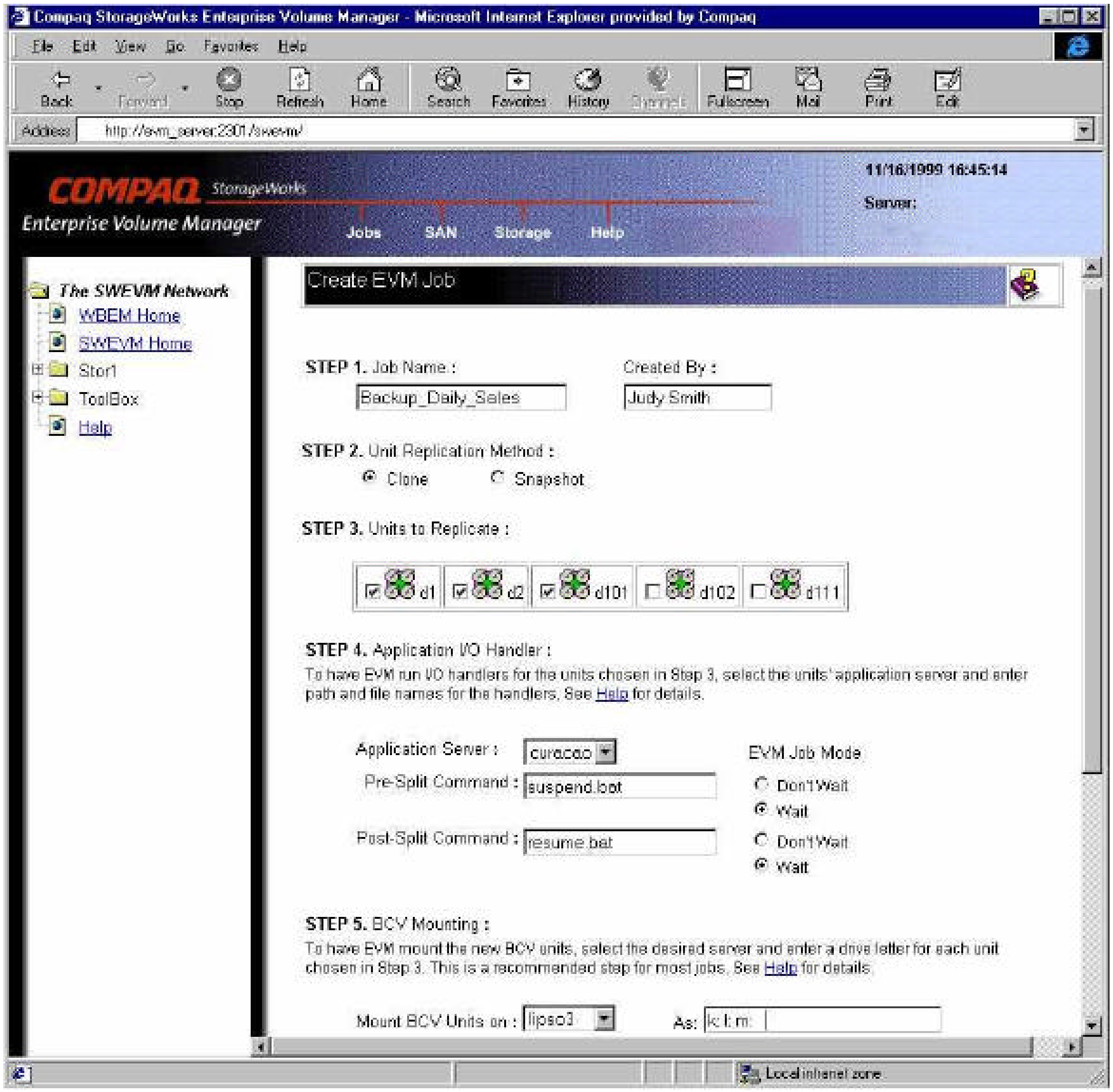
EVM is a web-based software tool that provides a method of creating instant point-in-time copies of RA8000- and ESA12000- based storage volumes. These copies, called BCVs, can be migrated to and mounted on any other supported host in the EVM storage network. One advantage of this feature to the Oracle DBA is the ability to perform back up operations on a copy of the production data and to do so on a completely different host. This offloads the sometimes significant CPU overhead associated with the back up process from the database server to a dedicated back up host.

Oracle’s online or hot back-up utility allows tablespaces to be backed up while the database is open. Although the online back-up feature eliminates the need to take the database offline before performing a back-up, there is a performance penalty associated with its use.

Also, it is necessary to leave each tablespace in hot back-up mode for as long as the back-up requires to complete, possibly several hours or more. With EVM, it is only necessary to leave the tablespace(s) in hot back-up mode for a few seconds while the BCVs are split from the production volumes. In short, EVM eliminates the performance penalty associated with online back-ups and expands the back-up window to 24 hours a day, 365 days a year.

In addition to facilitating back-up, EVM also provides the ability to dynamically mount copies of one or more database volumes. It will mount these database volumes on a separate host for batch processing, data mining, application testing, data warehousing or other processes that could benefit from having an independent copy of the database.

In order for EVM to be effective, it must ensure that the BCVs contain a valid copy of the database that is consistent at the block level. This involves ensuring that there are no dirty buffers and that the data remains valid during the brief period while the source volume and the BCV are being separated from each other. EVM accomplishes this through the use of the PRE and POST command fields in Step 4 of the Create Job page, as shown below.



Create EVM Job

STEP 1. Job Name : Backup_Daily_Sales

Created By : Judy Smith

STEP 2. Unit Replication Method :

● Clone ○ Snapshot

STEP 3. Units to Replicate :

☑ d1 ☑ d2 ☑ d101 □ d102 □ d111

STEP 4. Application I/O Handler :

To have EVM run I/O handlers for the units chosen in Step 3, select the units' application server and enter path and file names for the handlers. See Help for details.

Application Server : curacao

EVM Job Mode

Pre-Split Command : suspend.bat ○ Don't Wait ● Wait

Post-Split Command : resume.bat ○ Don't Wait ● Wait

STEP 5. BCV Mounting :

To have EVM mount the new BCV units, select the desired server and enter a drive letter for each unit chosen in Step 3. This is a recommended step for most jobs. See Help for details.

Mount BCV Units on : lipso3 As: k: l: m:

Figure 1: Create EVM Job Screen

These fields are used in conjunction with Oracle's online back-up feature to ensure that the BCVs contain a consistent copy of the database files.

The PRE and POST fields both accept a standard path\filename as input and are used to execute batch files, scripts or compiled executables on the specified application server at certain points during the execution of the EVM job. While the creation of these files is left up to the user, they are intended to be used to execute a short sequence of specific SQL commands which will be described later. The examples given below use the Oracle command line mode to execute short SQL scripts. There are many other ways of accomplishing this same goal and, if the results are the same, the details of the implementation are left up to the user.

EVM will execute the file contained in the PRE field just before the BCV is split from the source volume. This file should checkpoint the database and place the necessary tablespaces in online back-up mode. The POST file is executed just after the BCV is split and should be used to take each tablespace out of the online back-up mode. If used correctly, the PRE and POST files will ensure that the BCV contains a valid and consistent copy of the database.

More About Online Back Up

Oracle has a built-in feature that allows the execution of online or hot back-ups. This function allows back-up operations to take place while the database is running and users are continuing to access the database. This eliminates the need to shut down the database each time a back up is needed. Oracle's implementation of this feature is through the BACK UP parameter of the ALTER TABLESPACE command. For example:

```
ALTER TABLESPACE tablespace_name BEGIN BACK UP;
```

would place *tablespace_name* in online back-up mode. Similarly:

```
ALTER TABLESPACE tablespace_name END BACK UP;
```

would take *tablespace_name* out of online back-up mode.

Note: The unit of back-up for this function is the tablespace. In order to back up an entire database, it will be necessary to issue the above command for each tablespace.

Although online back-ups deal with individual tablespaces, it possible to back up several tablespaces at once by issuing multiple online back-up commands. Normally, this would not be recommended due to the undesirable side effects of having multiple tablespaces in online back-up mode simultaneously. Fortunately, EVM eliminates these problems by reducing the amount of time that the tablespaces need to be in online back-up to only a few seconds.

In order to use the online back-up feature, the database must be operated in ARCHIVELOG mode. Otherwise, online back-ups will not work. To enable archivelog mode, place the database in the mounted (but not open) state and issue the following commands from the Server Manager prompt:

```
ALTER DATABASE ARCHIVELOG;

ARCHIVE LOG START

ALTER DATABASE OPEN;
```

To make these changes permanent, you must modify your INIT.ORA file.

Comprehensive Oracle Backup Plan

A comprehensive Oracle back-up plan should include the data files, the archived redo log files and the control file. It is assumed that the Oracle program files are included as part of the OS back-up.

Data Files

Data files are the physical files that contain a tablespace's data and are the most critical components of any back up. By default, Oracle creates the SYSTEM and USER tablespaces and their associated data files. In addition to these default data files, each additional tablespace will contain one or more data files. As a database grows, new data files may be added, thereby complicating the back up plan. In order to develop an effective back up plan, it is important to understand where each data file is located and what tablespace it belongs to.

Most Oracle databases are comprised of multiple volumes. Since the unit of online back up is the tablespace; any EVM jobs you create must take into account that a single tablespace may span multiple volumes. If this is the case, it will be necessary to create a BCV of every volume that contains a portion of the tablespace. Let's say your Oracle implementation involves four volumes (E:, F:, G: and H:) and a total of four tablespaces (SYSTEM, USER, TEMP, and WEB). Assume the data files for these tablespaces are located as follows:

Volume E:

SYSTEMorc1.ora	Data file #1 for SYSTEM
USERorc1.ora	Data file #1 for USER
TEMPorc1.ora	Data file #1 for TEMP
WEBorc1.ora	Data file #1 for WEB

Volume F:

SYSTEMorc2.ora	Data file #2 for SYSTEM
USERorc2.ora	Data file #2 for USER
WEBorc2.ora	Data file #2 for WEB
WEBorc3.ora	Data file #3 for WEB

Volume G:

WEBorc4.ora	Data file #4 for WEB
WEBorc5.ora	Data file #5 for WEB

Volume H:

Archived redo logs only

In order to back up the WEB tablespace, an EVM job would need to create BCVs for E:, F:, and G:. This is because the WEB tablespace has data files located on all three volumes. The SYSTEM and USER tablespaces would need BCVs of both E: and F:, and TEMP would need a BCV of E: only.

When EVM creates a BCV, it will use the same number of disks that the source volume uses. This applies to both types of BCV (clone or snapshot). For example, if you want to create a BCV from a six-disk RAID3/5 set, you must have at least six unused disks of the same capacity available for the BCV.

From the example above, backing up the WEB tablespace will involve BCVs of three different source units. Since all three source units must be replicated within a single EVM job, this operation will clearly require a large number of disks. If there are not enough unused disks available, it will be necessary to either add more disks or to consolidate WEB's data files so that they occupy fewer volumes. Proper data file management can help prevent problems like this from occurring.

Archived Redo Logs

As mentioned earlier, the online back up function requires that the database be run in ARCHIVE LOG mode. In this mode, as each online redo log fills up, Oracle copies it to the location specified by the ARCHIVE_LOG_DEST parameter and begins filling the next online redo log. In this manner, the online redo logs are recycled and a sequential trail of redo is maintained so that any changes that are lost can be rolled forward easily. In order to restore a tablespace up to a point it time, it is necessary to maintain both the data file back-ups as well as back ups of all the archived redo logs that have been generated since the data files were backed up.

There are several methods of backing up archived redo logs. The method you choose will depends primarily on where the logs are located. If they reside on an HSG80-based volume, you can create a clone or snapshot BCV of the volume and back up the archived redo logs in the same manner as the data files.

Additional redo will be generated constantly, and it will be necessary to back these files up periodically. Remember that complete database recovery requires the most recent back-up of the data files plus all of the archived redo logs that have been generated since the data files were backed up. Any gap in the archived redo log sequence will prevent complete recovery, though it will be possible to roll the transactions forward to the point of the first missing log.

Once archived redo logs have been backed up they can be deleted to free up space. However, if you have enough free space, keeping the archived redo logs on disk between data file back ups can significantly speed the recovery process if only data files are lost.

Control File

The control file contains the schema of the database and, among other things, includes the data necessary to restore the database from an online back-up. However, this data is not written to the control file until after all tablespaces have been taken out of online back-up mode. Therefore, the control file cannot be backed up until after online back-up mode has ended. By that time, each BCV has already been split from its source volume so there is no simple way to get the current control file onto the BCV. Remember that any copies of the control file that are located on the BCV(s) are old copies in the sense that they do not contain the data needed to restore the database from the online back-up. If possible, it is best not to back up these files at all since restoring them will overwrite the current control file, the one containing the information needed to perform the recovery. Considering the problems that can result from improperly restoring a control file, many Oracle administrators prefer not to back it up at all. By default, the control file is mirrored to at least two locations, so the loss of one copy will not seriously impact database operation. Whether or not to back up the control file is in important decision and should be made carefully after fully considering the implications. Consult the Oracle documentation for more information on this subject.

Online Redo Logs

Online redo log files should never be backed up. While actually backing them up is not a problem, restoring them will overwrite the current online redo logs and will make it impossible to completely recover the database. Considering the dangers involved it is best never to back them up in the first place. As with the control file, it is possible to mirror the online redo logs to multiple locations so that the loss of one set will not adversely affect the operation of the database.

Restoring an Oracle Database That Has Been Backed Up Using EVM

Recovering a database that has been backed up using EVM involves two basic steps; restoring the lost files from tape and applying the redo logs.

Restoring lost files

In most cases, EVM can't add much value to this process since such an operation would involve first, creating a partitioned and formatted temporary volume; then restoring the data from tape to the temporary volume, migrating and mounting the volume to the Oracle server, and finally copying the data from the temp volume to the original source volume. Given the number of steps involved in this process, it is generally faster and easier to just restore the data from tape directly back to the original volume, even if this involves restoring over the network.

In some cases it may be necessary or beneficial to use EVM to restore lost files. In this case, the suggested procedure is to:

1. Create a new volume of sufficient size through either CLI or SWCC and make it exclusively visible to the back-up server by setting the appropriate access IDs, offsets, or both.
2. Reboot the back-up server, then mount and format the new volume.
3. Restore the data from tape to the new volume.
4. Make the new volume exclusively visible to the Oracle DB server.
5. Reboot the Oracle DB server and mount the new volume.
6. Manually copy the data from the new volume over to the original source volume.

Alternatively you may be able to restore from tape directly to the original volume by migrating it to the back-up server, restoring from tape, and migrating back. This saves the final temp-to-source copy that is described above.

Applying the Redo Logs

This step in the recovery process will depend on what is being restored and whether the database needs to be open during recovery. Oracle provides methods for performing *data file recovery*, *tablespace recovery,* and *database recovery*. Consult the Oracle documentation for more information on these functions.

Sample PRE and POST Files

The following scripts are provided as examples only. It is the user's responsibility to verify the effectiveness and suitability of these scripts for their intended application.

The suggested method of issuing the following SQL scripts is to use Oracle's SQLPLUS utility in command line mode. The command to execute the SQL script evm-pre.sql in Windows NT might look something like this:

```
c:\orant\bin\sqlplus "@evm-pre" > pre.log
```

This command specifies the path to the SQLPLUS utility, passes the evm-pre.sql script as an input file and redirects the output to a file called pre.log. Redirecting the output to a log file is a very important step as will be explained shortly. This command is an example of what you would enter in the PRE field of the EVM Create Job page.

The following sample scripts can be used as a starting point for developing your own. As an example, if you wanted to back up tablespaces USR and WEB, a suitable PRE script would look something like this:

```
 Sample PRE routine (evm-pre.sql)

****

connect internal/letmein;

archive log list;

alter tablespace USR begin back up;

alter tablespace WEB begin back up;

quit;

****
```

Let's look at this script line-by-line.

```
connect internal/letmein;
```

The first line connects to the database with the specified username and password, in this case the user is "internal" and the password is "letmein."

```
archive log list;
```

The next line issues the archive log command with the list parameter. The output of this command will look something like this:

```
Database log mode               Archive Mode

Automatic archival              Enabled

Archive destination             E:\archive

Oldest online log sequence      266

Next log sequence to archive    267

Current log sequence            267
```

The relevant piece of information here is the oldest online log sequence, which in this case became 266. By identifying the oldest online log sequence we are able to redirect the output from sqlplus to a log file. It will be necessary to retrieve this information later in order to determine which archived redo logs need to be backed up as part of the current online back-up.

```
alter tablespace USR begin back up;

alter tablespace WEB begin back up;
```

The third and fourth lines of the script are the commands that actually place the tablespaces USR and WEB into online back up mode.

```
quit;
```

The last line exits the script and returns control to EVM.

When this script completes, the specified tablespaces will have been placed into hot back-up mode, thereby ensuring that the data files are consistent. EVM is now able to split the BCV volumes from the sources. This takes a few seconds and, once done, the next step is to take the tablespaces out of online back-up mode through the use of the POST script.

Sample POST routine

```
****

connect internal/letmein;

alter tablespace USR end back up;

alter tablespace WEB end back up;

archive log list;

alter system switch logfile;

quit;

****
```

connect internal/letmein;

Here again, the first line connects to the database using the specified user ID and password.

```
alter tablespace USR end back up;

alter tablespace WEB end back up;
```

The second and third lines take the tablespaces USR and WEB out of online back up mode.

```
archive log list;
```

The fourth line is the same as it was in the PRE script except now the output will look something like this:

```
Database log mode               Archive Mode

Automatic archival              Enabled

Archive destination             E:\archive

Oldest online log sequence      312

Next log sequence to archive    313

Current log sequence            313
```

The relevant piece of information here is the current log sequence number in this case became 313. Once again, by identifying the current log sequence number we are able to change the output from sqlplus and redirect to a log file.

```
alter system switch logfile;
```

The next line of the script forces a logfile switch. This causes Oracle to stop recording redo in the current log sequence number and forces a switch to the next sequential log file; 314. The log file that was current (313) is then archived. The Oldest online log sequence identified in the PRE script and the Current log sequence from above now specify the entire range of archived redo logs that must be backed up in order to ensure a complete recovery. These logs and any additional archived redo logs that are generated after the BCVs were created must be backed up separately; either by another EVM job or outside of EVM altogether.

```
quit;
```

As before, the last line exits sqlplus and returns control to EVM.

When the POST script completes, the specified tablespaces will have been taken out of hot back up mode. EVM is now able to migrate the BCV volumes and mount them on the specified host where they can be backed up.

Conclusion

If used correctly, EVM can significantly increase the efficiency of your Oracle back-up, restoring and migration operations. As with any change to your database configuration or back up plan it is vitally important that you understand all aspects of your back-up and restore operations and test them thoroughly before implementing them in a production environment.