Maximizing hardware return on investment is an important business concern. To conserve hardware resources, many IT environments run multiple databases on a single server. With this proliferation of databases also comes the proliferation of database clones. Database clones are very useful for testing, developing, or keeping a point-in-time image for evaluation purposes. However, too many database clones take up a lot of server space.

When there is a shortage of server hardware, and when the source server has enough resources to run another database, the clone can run on the same server as the source database. For example, if needed, you could clone a development database back onto the development server.

Cloning a database onto the same server is more complicated than cloning the database onto a different server. To isolate the clone from its source database on the same server, you have to know which files to edit and how to edit them. Otherwise, any mistake that you might make when you create the clone might overwrite the source database and corrupt it. Not surprisingly, Oracle Database administrators are often reluctant to try to create clones on the same server. Even though doubling up would conserve hardware resources, sometimes the risk is not worth taking.

This document explains how to isolate a cloned database from its source. This document explains how to set up the cloned database using different storage locations from the source database. Initially, all of the pointers in the cloned database point to the original locations for the source database. You have to change the locations before you can use the cloned database. This document describes how to edit all of the references to the file systems so that it is safe to use the cloned database.

This document also describes a solution that simplifies cloning. Specifically, this document focuses on expediting the important task of cloning an Oracle Database to the same server. The solution combines Oracle Database products with IBM storage subsystems and the premium features, FlashCopy® and VolumeCopy, which come with both DS Storage Manager and IBM DS Storage Manager 10. This method of cloning a database requires less effort than using an Export/Import function or backing up and restoring the database using Oracle RMAN.
The intended reader for this document is an Oracle Database administrator with experience in the following areas:

- Oracle Database and its related components
- Cloning an Oracle Database
- Storage, including an understanding of data services and the premium features that support those services

Benefits of Using This Method

You can use FlashCopy and VolumeCopy with Oracle to create a copy of the database and clone it onto the same server as the source database. Then you can perform certain critical functions on the copy rather than on the production database. Using a copy for critical functions has many advantages:

- No disruption of the source production database
- Fewer staff resources
- Anytime duplication of the database
- Data replication for recovery of the production database

FlashCopy Consistency Groups

IBM DS5000, DS4000, and DS3000 support FlashCopy Consistency Groups. This feature allows the FlashCopy operation to be performed atomically on a group of up to 50 LUNs.

FlashCopy Consistency Groups apply to Oracle Database environments in addition to other applications. Oracle Database can use either filesystems or Oracle Automatic Storage Management (ASM) to exploit this feature. ASM could be configured to stripe database data across multiple LUNs. ASM operates independently from the Oracle database. While it is still possible to quiesce the database instance, the ASM instance cannot be quiesced. ASM runs background tasks to rebalance data on addition or removal of LUNs, and these tasks cannot be quiesced. To guarantee that the FlashCopy replica can be used for backups or for cloning, a FlashCopy replica has to be taken across all the LUNs atomically, at the same instance, while all I/O is suspended at the array level. This is where FlashCopy Consistency Groups come in. This FlashCopy Consistency Groups capability can also be used for databases using filesystems over multiple LUNs or even for databases that have a mixture of both ASM and filesystems.
Benefits of Using FlashCopy

The FlashCopy premium feature lets you create a point-in-time copy of data on a logical drive. This copy is the logical equivalent of data on the complete base logical drive, but the point-in-time copy is created much faster than a complete physical copy, typically, within a few seconds. Due to the architecture of FlashCopy, creating the copy typically requires much less disk drive space. FlashCopy logical drives become a read/write logical drive that can be assigned to the same host or to a secondary host depending on the content of the logical drive. You can use the FlashCopy logical drives for many purposes, such as application backups, while leaving the base logical drive online and accessible to the user.

Benefits of Using VolumeCopy

The VolumeCopy premium feature is an in-box mechanism for replicating logical drive data within the same storage subsystem. VolumeCopy creates a complete physical copy of a source logical drive to another target logical drive within the same storage subsystem. When you use VolumeCopy with FlashCopy, the replication occurs while the source logical drives remain online and accessible to the user.

Setting Up the System Environment

Use the reference architectures as guidelines for setting up your own system environment. The reference architectures serve as examples only. You can modify the system environment as necessary to suit your needs. There are two reference architectures:

- The DS4000™ reference architecture consists of one primary sever and one secondary server attached to two DS4000 controllers. See Figure 1 on page 6.
- The DS5000™ reference architecture consists of one primary server and one secondary server attached to one DS5000 controller and one DS4000 controller. See Figure 2 on page 7.

The reference architectures were developed for testing seven common scenarios that an Oracle Database administrator would regularly encounter. This document describes only one of those scenarios. The other six scenarios are described in separate documents. Those documents are listed in “Appendix A: References.”

The reference architectures use three premium features—VolumeCopy, FlashCopy®, and Enhanced Remote Mirroring (RM)—to test various backup and recovery methods and to test multiple development and upgrade scenarios. The specific Oracle scenario described in this document might not use all three premium features.
Technical Specifications

Here are the specifications for components of the reference architectures.

Operating System Information

Red Hat Enterprise Linux 5.1 Version 2.6.18-53.el5
The RDAC driver is MPP Driver Version 99.03.C000.0005

Oracle Information

Oracle Database EE 10g Release 2 Version 10.2.0.3
Oracle Database EE 11g Release 1 Version 11.1.0.6
Storage Information

DS4800
  Firmware Version 07.10.22.00
  NVSRAM Version N1815D48R1010V05

DS5100
  Firmware version 07.30.22.00
  NVSRAM version N1818D51R1030V09

IBM DS Storage Manager 10 Version 10.10.G5.11

Logical Drive Configuration

For each reference architecture, the primary storage subsystem has the following logical drives:

- Oracle distribution logical drive /u01
- Primary Oracle Database logical drives
- FlashCopy logical drives
- Oracle VolumeCopy logical drives

For each reference architecture, the secondary storage subsystem has the following logical drives:

- FlashCopy logical drives
- Secondary RM logical drives for the Oracle distribution and database logical drives
Reference Architecture

Figure 1 shows that the DS4000 reference architecture consists of one primary server and one secondary server attached to two DS4800 controllers.

Figure 2 on page 7 shows that the DS5000 reference architecture consists of one primary server and one secondary server attached to one DS5100 controller and one DS4800 controller.
Figure 2 DS5000 Reference Architecture
Logical Drive Configuration

Figure 3 shows the logical drive configuration in the tests of IBM DS Storage Manager 10 premium features. Both reference architectures use the same logical drive configuration.

Figure 3  Logical Drive Configuration
Setting Up the Premium Features

After the system environment is set up as you want, then you must set up the premium features. Enable both the FlashCopy premium feature and the RM premium feature. You can enable the premium features from either host. You must enable a premium feature only once for the feature to become fully enabled throughout the storage subsystem. To enable the premium feature, obtain the Feature Key file from IBM.

Enabling a Premium Feature

1. In the IBM DS Storage Manager 10, open the Subsystem Management Window for the storage subsystem that will contain Oracle data.
2. From the main menu, select Storage Subsystem >> Premium Features >> Enable.
3. Navigate to the *.KEY file that contains the activation key for the premium feature. When IBM sent the *.KEY file, you specified its file name and location.
4. Select the file, and then click OK.
5. Repeat this procedure to activate an additional premium feature key.

Setting Up FlashCopy

FlashCopy requires a repository logical drive to store data that changes from the original data on the base logical drive. FlashCopy is a point-in-time copy, not a complete replication. The FlashCopy process reads data from the base logical drives. When data is rewritten on the base logical drive, the original data is copied to the FlashCopy repository to retain the point-in-time copy. The repository also contains any changes that the FlashCopy replica incurs after being mapped to a host, because that host can perform writes to the FlashCopy logical drive.

When you use a FlashCopy replica against the primary application logical drives, you must consider both the proper sizing of the FlashCopy logical drive and the potential impact on performance. The default size for a logical drive is 20 percent. However, the optimal size of the repository logical drive depends on the application and on how much data changes during the time that the FlashCopy remains active.

- When the application changes very little data, and when the FlashCopy operation is performed for backup, the repository can be relatively small.
- When the application routinely changes the existing data, and when the FlashCopy process is used for testing or upgrading purposes, the FlashCopy repository must be relatively large—up to 100 percent of the base logical drive.
Setting Up VolumeCopy

VolumeCopy creates an identical copy of the source logical drives within the same storage subsystem. VolumeCopy requires target logical drives to be the same size as the source logical drive. VolumeCopy turns the source logical drive into a read-only logical drive. Therefore, you must use VolumeCopy with FlashCopy on production logical drives. Typically, you take a FlashCopy of a logical drive, and then start a logical drive copy of the FlashCopy replica.

Detailed Example Using This Method

After you set up the premium features, log on to Oracle to complete the procedures. This section gives a high-level overview of the work that you must do in Oracle. The following sections provide more detailed information.

In the test case, the source database resides on HOST1, named pfdb2. The VolumeCopy procedure copies all of the pfdb2 data files and the Oracle binary files. The new database is cloned to the same host and is named pfdb3. The source database pfdb2 uses Oracle binary files and file systems on mount point /u01, and the new database uses mount point /u02 for all the data files and Oracle binary files.

NOTE  When you want to clone a database that uses Oracle Automatic Storage Management (Oracle ASM), you cannot mount a copy of the ASM disks on the same server as the source database. You can use VolumeCopy to clone an ASM database, but you must mount the ASM disks on another server that is not using Oracle ASM.

Perform a FlashCopy and a VolumeCopy to create a clone of mount point /u01. The mount point /u01 contains the binary files and all of the data files of the pfdb2 database. When your database has more mount points, make sure that you include all of the mount points in the FlashCopy process and the VolumeCopy process. Mount this VolumeCopy as /u02 on HOST1.

Creating a FlashCopy and a VolumeCopy

VolumeCopy marks the source as read-only. Therefore in production environments, you must use FlashCopy with VolumeCopy. When you use FlashCopy with VolumeCopy, the source data remains read/write. Create the initial FlashCopy replica before you quiesce the database to expedite quick FlashCopy because the initial creation causes the LUN to be initialized. Subsequent use of the disable / re-create FlashCopy is extremely quick.

1 Quiesce the database in preparation for a FlashCopy and VolumeCopy. Enter the following commands.

```
SQL> alter system switch logfile;
SQL> alter database begin backup;
```
SQL> alter system quiesce restricted;
SQL> select active_state from v$instance;

The following response appears.

<table>
<thead>
<tr>
<th>ACTIVE_STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUIESCED</td>
</tr>
</tbody>
</table>

2 Start a flashcopy re-create using IBM DS Storage Manager 10.

Right-click the logical drive and select **Re-create FlashCopy**. Use a repository logical drive size of 20 percent.

3 Bring the source database back online. Enter the following commands.

SQL> alter system unquiesce
SQL> alter database end backup;
SQL> alter system switch logfile;

The following response appears.

Database altered.

**NOTE** You might need this online log file to complete the recovery of the database.

4 Use the newly created FlashCopy logical drive as the source to create a VolumeCopy target that uses IBM System Storage DS Storage Manager.

After the VolumeCopy completes, you can remove the FlashCopy logical drive, and then you can map the VolumeCopy logical drives to the host.

5 Map the target VolumeCopy and make the target visible to the operating system.

a  Run `mppBusRescan` to locate new devices.

b  Run `SMdevices` to make sure that the operating system can find the new devices.

c  Make a mount point with `mkdir /u02`.

d  Mount the file system as `/u02`. 
Cloning the Database

To clone the database, you must prepare both the source database and the target location. You must prepare the source database first.

**CAUTION**  Possible database corruption - You are likely to corrupt another database on the server if you try to start the cloned database on the same server as the source database without first changing all of the database references:
- Data files
- Log files
- Binary Locations

Preparing the Source Database to Be Cloned

In the test case, the source database was pfdb2.

1. Clone the source database and change its name.

   Rebuild the controlfile using the new data file locations and the new mount points. First back up the controlfile in the source database to a text file, and then edit the controlfile. Enter the following command.

   ```sql
   SQL> alter database backup controlfile to trace as '/home/oracle/pfdb2ctrlfile.ora' reuse;
   ```

   The following response appears.

   ```
   Database altered.
   ```

2. Find out where the data objects are located. Enter the following commands.

   ```sql
   SQL>  spool '/home/oracle/gatherpfdb2_info.txt'
   SQL>  col name format a30
   SQL>  col value format a60
   SQL>  select name, value from v$parameter where value like '%$/u01%';
   ```
The following response appears.

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>spfile</td>
<td>/u01/app/oracle/product/10g/dbs/spfilepfdb2.ora</td>
</tr>
<tr>
<td>control_files</td>
<td>/u01/oradata/pfdb2/control01.ctl,</td>
</tr>
<tr>
<td></td>
<td>/u01/oradata/pfdb2/control02.ctl,</td>
</tr>
<tr>
<td></td>
<td>/u01/oradata/pfdb2/control03.ctl</td>
</tr>
<tr>
<td>db_recovery_file_dest</td>
<td>/u01/oradata/pfdb2/fra</td>
</tr>
<tr>
<td>background_dump_dest</td>
<td>/u01/app/oracle/admin/pfdb2/bdump</td>
</tr>
<tr>
<td>user_dump_dest</td>
<td>/u01/app/oracle/admin/pfdb2/udump</td>
</tr>
<tr>
<td>core_dump_dest</td>
<td>/u01/app/oracle/admin/pfdb2/cdump</td>
</tr>
<tr>
<td>audit_file_dest</td>
<td>/u01/app/oracle/admin/pfdb2/adump</td>
</tr>
<tr>
<td>dg_broker_config_file1</td>
<td>/u01/app/oracle/product/10g/dbs/dr1pfdb2.dat</td>
</tr>
<tr>
<td>dg_broker_config_file2</td>
<td>/u01/app/oracle/product/10g/dbs/dr2pfdb2.dat</td>
</tr>
<tr>
<td></td>
<td>9 rows selected.</td>
</tr>
</tbody>
</table>

Enter the following command.

SQL> show parameter dispatchers

The following response appears.

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>dispatchers</td>
<td>string</td>
<td>(PROTOCOL=TCP)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(SERVICE=pfdb2XDB)</td>
</tr>
</tbody>
</table>
Enter the following commands.

SQL> col name format a60
SQL> select name from v$datafile;

The following response appears.

NAME
------------------------------------------------------------
/u01/oradata/pfdb2/system01.dbf
/u01/oradata/pfdb2/undotbs01.dbf
/u01/oradata/pfdb2/sysaux01.dbf
/u01/oradata/pfdb2/users01.dbf
/u01/oradata/pfdb2/example01.dbf
/u01/oradata/pfdb2/test_tts.dbf
6 rows selected.

Enter the following commands.

SQL> col member format a80
SQL> select member from v$logfile;

The follow response appears.

MEMBER
------------------------------------------------------------
/u01/oradata/pfdb2/pfdb2/redo03.log
/u01/oradata/pfdb2/pfdb2/redo02.log
/u01/oradata/pfdb2/pfdb2/redo01.log
3

Create the pfile for the new database.

a Create a new pfile to use initially when you start the database to create the controlfile.

   SQL> create pfile='/home/oracle/initpfdb3Temp.ora' from spfile;
   SQL> spool off

b Exit sqlplus.

4 Delete all of the control files on the newly cloned database.

   Check the gatherpfdb2_info.txt file that you created in step 1 for the locations of the control files, and then delete them.
5 Gather the information that you must edit at the operating system level to make
the files and directories unique. On the new mount point, the directories point to
pfdb2, which is the source database. Enter these commands in the following
order.

   a cd /u02
   b find . -name *pfdb2*
   c find . -name *PFDB2*

6 You must edit the information that you gathered from the two find commands in
the previous step.

   7 Edit the directory names to reference the new database name pfdb3.
   7 Edit the content of the files.
   7 Ignore the trc files, log files, and so forth.

7 Change the name of a directory from the source name to the target name.

In the test case, the name was changed from pfdb2 to pfdb3 using the following
command.

    mv /u02/oradata/pfdb2 /u02/oradata/pfdb3

8 Edit all of the files that you found.

   a Rename the files to the new name.
   b Change the content of the file to reference the new mount point and database
      name.

---

**IMPORTANT** Make sure that there is no reference to pfdb2 either in any files
listed under the new mount point or in the contents of any files on
the new mount point.

---

9 Change all of the Oracle environment variables. Point the environment variables to
the new SID pfdb3 along with the new locations for ORACLE_HOME,
ORACLE_BASE, PATH, and any other variable that points back to the old mount
point or the source database pfdb2.

---

**IMPORTANT** Make sure that all of the Oracle environment variables are pointing
to the new locations and new database by issuing env on the
command prompt.
Detailed Example Using This Method

10 Edit the pfile.

- Copy the pfile that you created from the source database in step 3 to the new $ORACLE_HOME/dbs directory. In the test case, the pfile was named initpfdb3Temp.ora.
- Edit this pfile to change all references of the old mount point to the new mount point along with changing all references of pfdb2 to pfdb3.
- Save your changes.

11 Edit the controlfile that you created earlier (in step 1) from the source database with the following changes.

- Change all of the references of the old mount point to the new mount point.
- Change any references of pfdb2 to pfdb3.
- Edit the pfile entry on the STARTUP NOMOUNT command to reference the new pfile in the $ORACLE_HOME/dbs directory.
- Change the create controlfile statement to say set database instead of use database.
- Make sure that the Create Controlfile statement specifies RESETLOGS.
- Delete all of the lines from the beginning to the controlfile file down to the Startup Nomount statement in the section that specify RESETLOGS.

Preparing the Target Database on the Same Server

Change the environment settings, and then complete the following steps. In the test case, the target database was pfdb3.

1 Create a control file for the newly-cloned database.

Open one window for editing the controlfile and one window for editing sqlplus. Then copy the commands from the controlfile window and paste them into the sqlplus window. Having both windows open lets you view each step of the cloning process and catch any errors that might arise. Alternatively, you can script this procedure and execute it from sqlplus. To execute the script, enter the following commands.

```
oracle@HOST1$ sqlplus
SQL> conn / as sysdba
SQL> STARTUP NOMOUNT pfile='/u02/app/oracle/admin/pfdb3/scripts/initpfdb3Temp.ora'
SQL> CREATE CONTROLFILE set DATABASE "pfdb3" RESETLOGS ARCHIVELOG MAXLOGFILES 16 MAXLOGMEMBERS 3 MAXDATAFILES 100 MAXINSTANCES 8 MAXLOGHISTORY 292
```
LOGFILE
  GROUP 1 '/u02/oradata/pfdb3/pfdb3/redo01.log'  SIZE 50M,
  GROUP 2 '/u02/oradata/pfdb3/pfdb3/redo02.log'  SIZE 50M,
  GROUP 3 '/u02/oradata/pfdb3/pfdb3/redo03.log'  SIZE 50M
-- STANDBY LOGFILE
DATAFILE
  '/u02/oradata/pfdb3/system01.dbf',
  '/u02/oradata/pfdb3/undotbs01.dbf',
  '/u02/oradata/pfdb3/sysaux01.dbf',
  '/u02/oradata/pfdb3/users01.dbf',
  '/u02/oradata/pfdb3/example01.dbf',
  '/u02/oradata/pfdb3/test_tts.dbf'
CHARACTER SET WE8ISO8859P1

2 Query the mounted database to make sure that all destinations now point to the
   correct mount point and directories.
   SQL> col name format a30
   SQL> col value format a60
   SQL> select name, value from v$parameter where value like '%/u01%';

3 Create a new spfile for the new database.
   SQL> create spfile from pfile='/u02/app/oracle/admin/pfdb3/
scripts/initpfdb3Temp.ora';

4 Shut down the database to make sure that you can bring it up successfully.
   SQL>  Shutdown immediate;
   SQL>  STARTUP MOUNT

5 Edit the service_names parameters and the dispatchers parameters to the
   new values.
   SQL>  Alter system set service_names =’pfdb3.domainname’
   scope=both;
   SQL>  Alter system set dispatchers=’(PROTOCOL=TCP)
   (SERVICE=pfdb3XDB)’ scope=both;

6 Open the database.
   SQL>  ALTER DATABASE OPEN resetlogs;

If the database needs more recovery before it can be opened, take the following
steps.
   a Find the location of the logfile that was created by the switch logfile
      command after the FlashCopy process.
   b Issue a recover database command using backup controlfile.
   c Enter the path and name of the logfile in step a.
d  Wait to receive the media recovery complete message.

e  Enter the following command.
    alter database open resetlogs

7  Add the temporary tablespace definitions. The following code is from the test case.

    SQL> ALTER TABLESPACE TEMP ADD TEMPFILE '/u02/oradata/pfdb3/temp01.dbf' SIZE 20971520 REUSE AUTOEXTEND ON NEXT 655360 MAXSIZE 32767M;

    Your TEMPFILE specifications will be different. Refer to the controlfile text file that was created in step 1 for the exact syntax.

8  Verify that the database is up and open for use. Run the following query.

    SQL> select instance_name, host_name, database_status from v$instance;

    The following response appears.

    INSTANCE_NAME  HOST_NAME  DATABASE_STATUS
                   -----------  -----------  ---------------
                   pfdb3        HOST1        ACTIVE

9  (Optional) If you intend to access this database from outside of the server, you must edit the existing Listener.ora file to include the information about this new database.

    NOTE  To automatically start and stop this instance when invoking the Oracle dbstart and dbshut scripts, add an entry in the ORATAB file that specifies the following:
          • The new instance
          • The Oracle home
          • Whether you want to start up and shut down the instance automatically
Conclusion

After completing these procedures, you have an operational database that is a point-in-time clone of the source database on the same server. You can use this clone database for backup, application testing, migrations, upgrades, and other database manipulations. However you intend to use your database clone, you know that you have used your hardware resources wisely by locating the clone onto the same server as the source database.

If you prefer to have professional consultants service your cloning process, see “Contact Information.”

Contact Information

For more information, please visit the IBM web site at:

http://www.ibm.com
Appendix A: References

This document is one of seven detailed examples that explain how to complete common but important tasks often required from an Oracle database administrator. Refer to the other documents as needed.

- Cloning an Oracle Database to the Same Server Using FlashCopy and VolumeCopy on DS3400, DS4000, and DS5000
- Cloning an Oracle Database Using Enhanced Remote Mirroring on DS4000 and DS5000
- Forward Recovery of an Oracle Database Using Enhanced Remote Mirroring on DS4000 and DS5000
- Migrating to Oracle 11g Using Enhanced Remote Mirroring with Transportable Tablespaces on DS4000 and DS5000
- Recovering from Catastrophic Failures Using Enhanced Remote Mirroring for Data Replication on DS4000 and DS5000
- Safely Upgrading an Oracle Database Using Enhanced Remote Mirroring on DS4000 and DS5000
- Selective Restores Using Enhanced Remote Mirroring with Oracle Flashback Database on DS4000 and DS5000
Trademarks and Special Notices

© Copyright IBM Corporation 2009. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.