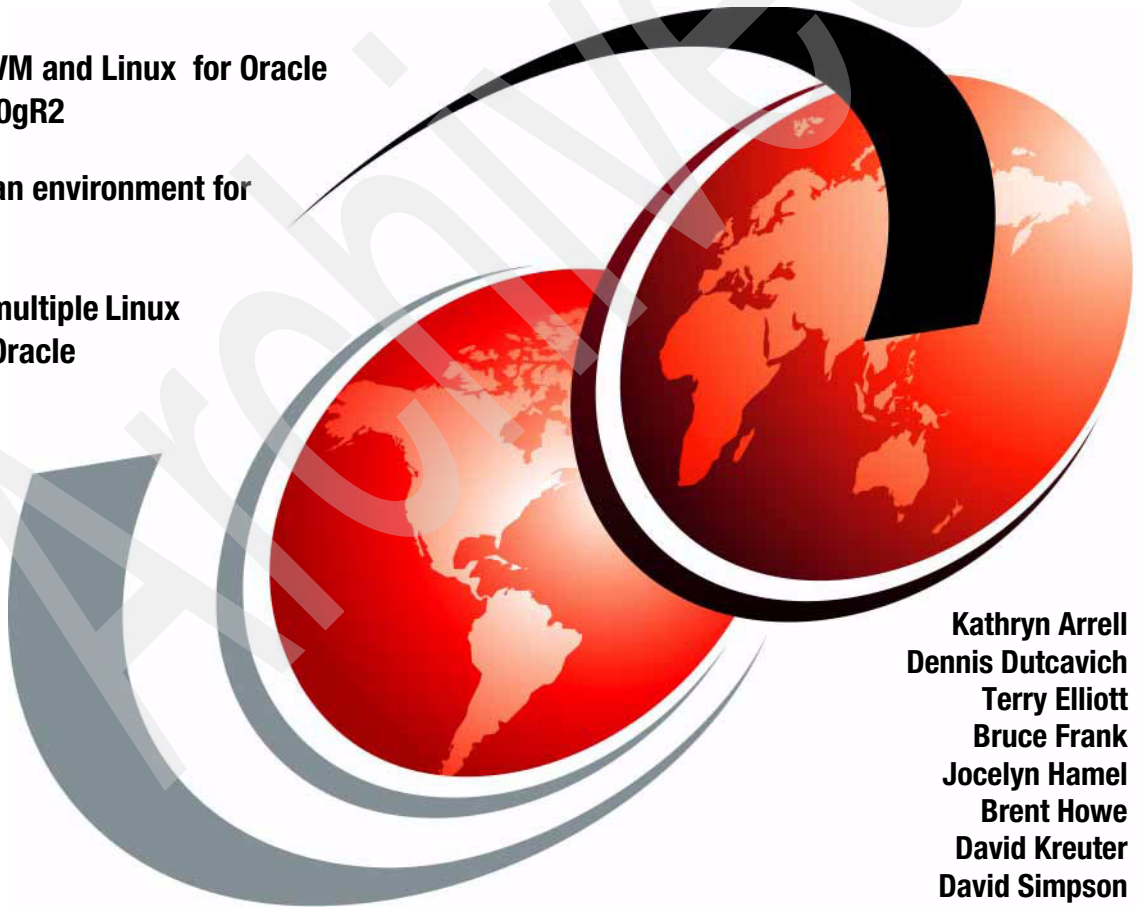
IBM

Using Oracle Solutions on Linux for System z

- Setting up VM and Linux for Oracle Database 10gR2
- Setting up an environment for Data Guard
- Setting up multiple Linux guests for Oracle



Kathryn Arrell
Dennis Dutcavich
Terry Elliott
Bruce Frank
Jocelyn Hamel
Brent Howe
David Kreuter
David Simpson

Redbooks

ibm.com/redbooks

International Technical Support Organization

Using Oracle Solutions on Linux for System z

April 2008

SG24-7573-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xv.

First Edition (April 2008)

This edition applies to Oracle Database 10gR2 running on SLES9, SLES10, and Red Hat 4.

Contents

Figures

Examples

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Redbooks (logo)  ®	ECKD™	RACF®
z/OS®	ESCON®	REXX™
z/VM®	FlashCopy®	System x™
zSeries®	FICON®	System z™
z9™	HiperSockets™	System z9®
AIX®	IBM®	WebSphere®
DS8000™	Redbooks®	

The following terms are trademarks of other companies:

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

Java, JDK, J2EE, Ultra, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries 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.

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

Preface

Linux® on System z™ offers many advantages to customers who rely upon the IBM® mainframe systems to run their businesses. Linux for System z takes advantage of the qualities of service in the System z hardware and in z/VM®, making it a robust industrial strength Linux. This provides an excellent platform for hosting Oracle® solutions that run in your enterprise.

This IBM Redbooks® publication describes experiences gained while installing and testing several Oracle features and solutions, such as:

- Oracle Data Guard
- IBM WebSphere® Application Server connecting to an Oracle database on Linux for System z
- Oracle BIEE connecting to an Oracle database on Linux for System z
- Oracle Data Pump

It also includes many general hints and tips on running Oracle products on IBM System z with Linux and z/VM.

The audience for this book includes database consultants, installers, administrators, and system programmers. This is not meant to replace Oracle documentation. It is just our experiences with installing and using Oracle products.

The team that wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Kathryn Arrell is an Oracle Specialist at the IBM/Oracle International Competency Center at IBM San Mateo. Previously she worked as an ERP specialist at the ITSO in Poughkeepsie, New York.

Dennis Dutcavich is a System z Oracle Specialist in IBM Sales and Distribution in the Americas. He supports Linux for System z presales opportunities.

Terry Elliott is a IBM System z Specialist working in the IBM/Oracle International Competency Center at IBM San Mateo. He has over 30 years of experience in information technology. Before joining the Center, Terry was as an ERP System z Performance Specialist.

Bruce Frank is an IBM System z Oracle Specialist in the IBM/Oracle International Competency Center at IBM San Mateo.

Jocelyn Hamel is System z IT Architect (zITA) working for IBM Canada.

Brent Howe is an IBM System z Oracle/Siebel® Specialist in the IBM/Oracle International Competency Center at IBM San Mateo.

David Kreuter is a System z consultant. He has over 25 years of large system experience. Through his consultancy, VM RESOURCES LTD, David provides consulting and training in z/VM, Linux, and networking.

David Simpson is a System z Oracle Specialist working for Advanced Technical Support (ATS) in the America's Sales and Distribution team. David previously worked for IBM Global Business Services hosting Oracle systems for consulting engagements.

Thanks to the following people for their contributions to this project:

Roy Costa
Mike Ebbers
Richard Conway
Mike Connolly
Bob Haimowitz
Lydia Parziale
International Technical Support Organization, Poughkeepsie Center

Tom Russell
IBM Canada

Rick Mccutchen
Sam Amsavelu
Richard Lewis
Mark Trbojevic
Pat Blaney
IBM USA

Guillaume Lasmayous
Olivier Manet
IBM France

Kei Kitamura
Michiyasu Takada
IBM Japan

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

`ibm.com/redbooks/residencies.html`

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  `ibm.com/redbooks`

- Send your comments in an e-mail to:

  `redbooks@us.ibm.com`

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Part 1

Installing Oracle Solutions

In this part, we provide a list of Oracle Solutions that run on Linux for System z at the time of writing of this book. This part also includes a summary of hints and tips on how to set up the Linux and VM environment for Oracle products. It has some detailed checklists on what to set up.

There are other IBM Redbooks publications where we have previously described in detail our experiences installing Oracle products on IBM System z:

- For Oracle 9i: *Experiences with Oracle for Linux on zSeries*, SG24-6552
- For Oracle DB 10gR1: *Experiences with Oracle 10g Database for Linux on zSeries*, SG24-6482
- For Oracle DB 10gR2, AS10g, and mixed configuration of EBS R11: *Experiences with Oracle 10gR2 Solutions on Linux for IBM System z,* SG24-7191

1

Overview of Oracle Solutions on Linux for System z

This chapter is an overview of Oracle Solutions that run on System z. Oracle has three main families of products:

- Oracle Database Server (9i, 10g, 11g)
- Oracle Fusion Middleware:
  - Oracle Application Server (9iAS and AS10g)
  - Oracle Identity Manager
  - Oracle Service-Oriented Architecture (SOA)
  - Oracle Collaboration Suite
  - Oracle Business Intelligence Enterprise Edition
- Oracle Application Suites:
  - Oracle 11i E-Business Suite
  - Oracle's PeopleSoft® Enterprise
  - Oracle's Siebel Applications

This chapter provides a brief overview of these solutions. See the following Web site for more detailed information:

http://www.oracle.com

1.1 Oracle Database Server (9i, 10gR1, 10gR2, and 11g)

On the IBM System z platform, Oracle delivered their production level Oracle9i Enterprise Edition Database Server in 2002, Oracle 10gR1 in 2004, and 10gR2 in 2006. Oracle 11g is projected for 2008.

Over the past four years, many clients have moved or added new databases workloads to run on Linux on the IBM System z platform. The i in 9i stands for Internet and the g in 10g stands for grid.

These releases are the same Oracle database that runs on the other platforms, and they include most of the components of the Enterprise Edition, including:

- Oracle Real Application Clusters
- Oracle OLAP
- Oracle Spatial
- Oracle Label Security
- Oracle Partitioning
- Oracle Data Mining
- Oracle Advanced Security

This includes the client package, which contains Oracle Net Services, Oracle Call Interface (OCI), and the Pro*C/C++ precompiler. The release notes indicate if a feature is not supported.

These products provide a complete Oracle9i database or Oracle 10g Database Product for development, testing, and production in Linux for System z.

1.2 Oracle Fusion Middleware

Oracle Fusion Middleware is the brand name for Oracle's middleware products. It consolidates Oracle Application Server 10g and related AS products and options, Oracle Data Hub, and Oracle Collaboration Suite into one family of infrastructure products.

Oracle Fusion Middleware is used as the middleware infrastructure to unify Oracle's application portfolio as part of Oracle's “Project Fusion.” Clients can use Oracle Fusion Middleware today for their custom applications.

For IBM System z, Oracle AS10g and Oracle Collaboration Suite (in a split configuration with the Database tier) run on Linux for System z today.

1.2.1 Oracle Application Server (9iAS and AS 10g)

In the third quarter of 2006, Oracle certified AS10g to run on Linux for System z. Oracle 9iAS does not run on Linux for System z.

The versions that were ported to Linux for System z in 2006 are:

- Oracle Application Server 10g Release 2 (10.1.2)
- Oracle Application Server 10g Release 3 (10.1.3)
- Oracle Application Server 10g Release 4 (10.1.4 for Identity Manager (IM))
- Oracle Application Server 10g Release x (10.1.3.1 for Service-oriented architecture (SOA))

1.2.2 Oracle Application Server Components of 10.1.2

As of December 2007, the components of 10.1.2 are:

- Oracle Application Development Framework (ADF) 10.1.2.1.0
- Oracle Application Server Metadata Repository Creation Assistant 10.1.2.0.3
- Oracle Application Server Certificate Authority 10.1.2.1.0
- Oracle Application Server Integration 10.1.2.0.2
- Oracle Application Server Single Sign-On 10.1.2.1.0
- Oracle BPEL Process Manager 10.1.2.0.2
- Oracle Business Intelligence Discoverer 10.1.2.1.0
- Oracle Containers for J2EE™ (OC4J) 10.1.2.0.2
- Oracle Content Management Software Developer Kit (SDK) 9.0.4.2
- Oracle Enterprise Manager 10g Application Server Control 10.1.2.1.0
- Oracle Forms 10.1.2.0.2
- Oracle HTTP Server 1.3.31
- Oracle HTTP Server 2.0.52
- Oracle Internet Directory 10.1.2.1.0
- Oracle Personalization 10.1.2.0.2
- Oracle Portal 10.1.2.1.0
- Oracle Portal Upgrade 10.1.4.0.0
- Oracle Reports 10.1.2.0.2
- Oracle Sensor Edge Server 10.1.2.0.2

- Oracle TopLink® 10g (9.0.4.5)
- Oracle Ultra™ Search 10.1.0.4.2
- Oracle Web Cache 10.1.2.1.0
- Oracle Wireless 10.1.2.1.0
- Oracle Workflow 2.6.3.5
- Oracle XML Developer Kit (XDK) 10.1.2.0.2

1.2.3 Oracle Application Server Components of 10.1.3

As of December 2007, the components of 10.1.3 are:

- Oracle Application Development Framework (ADF) 10.1.3.0.0
- Oracle Containers for J2EE (OC4J) 10.1.3.0.0
- Oracle Enterprise Manager 10g Application Server Control 10.1.3.0.0
- Oracle HTTP Server 2.0.52
- Oracle HTTP Server 1.3.34
- Oracle Sensor Edge Server 10.1.3.0.0
- Oracle TopLink 10g (10.1.3.0.0)
- Oracle XML Developer Kit (XDK) 10.1.3.0.0

1.2.4 Oracle Collaboration Suite

Oracle Collaboration Suite is a complete collaboration suite, including calendar, e-mail, files, voicemail, and workflow. It enables the consolidation of messaging and collaboration infrastructure to achieve cost efficiency. Oracle Collaboration Suite is supported in a split configuration mode where you can run the database server on Linux for System z.

1.2.5 Oracle Business Intelligence Enterprise Edition

Oracle BIEE can run with the Oracle database on Linux for System z. The Oracle BIEE application runs on a middle tier platform such as AIX® or x86.

1.3 Oracle applications running on Linux for System z

You can run three Oracle application solutions on Linux for System z. At present, it is the database tier that you can run in a mixed configuration mode with the Oracle database on Linux for System z for:

- Oracle E-Business Suite
- Oracle's PeopleSoft Enterprise
- Oracle's Siebel Applications

In Chapter 7, “Running Oracle EBS 11.5.10.2 split configuration with DB on Linux for System z “, in *Experiences with Oracle?® 10gR2 Solutions on Linux for IBM System z*, SG24-7191, we discuss our experience with Oracle E-Business Suite.

More information about Oracle MetaLink can be found at:

http://metalink.oracle.com/

1.4 Linux distributions for Oracle Solutions on System z

- Oracle9i is a 31-bit product. Oracle9i for Linux for System z runs on Novell SUSE Linux Enterprise Server 8 (SLES8) 31-bit and SLES8 64-bit. Oracle9i single instance is certified on Novell SUSE Linux Enterprise Server 9 (SLES9) 64-bit.
- Oracle 10g is a 64-bit product. Oracle 10g R1 and R2 run on SLES8 64-bit, SLES9 64-bit, and Red Hat AS4. For Red Hat, the minimum level of 10gR1 is 10.1.0.5 and for 10gR2 is 10.2.0.2. Oracle 10gR2 (10.2.0.3) is supported on SLES10.
- You can obtain the latest information about supported platforms at the following Web site:

  http://www.otn.oracle.com/support/metalink/content.html

1.5 Web sites for Oracle Solutions on Linux for System z

- You can purchase Oracle Products, including Oracle support licenses, through the Oracle Store, at http://store.oracle.com/.
- You can download CD images from the Oracle Technical Network (otn) at http://otn.oracle.com/. If you download from otn, the products do not automatically come with a support license. You should check with Oracle for your support coverage.

- Technical documentation is available at http://otn.oracle.com/.
- Support is available at http://metalink.oracle.com.

2

Preparing to install Oracle products on Linux for System z

This chapter provides current information about installing Oracle products on IBM System z. The hints and tips are based on commonly asked questions or problems. These include:

- Ensuring the Linux RPMs that Oracle requires are all installed.
- Ensuring the correct Oracle patches and patchsets are installed.
- Ensuring Linux and VM have sufficient resources to run Oracle products.

This chapter covers the following:

- Setting up VM and Oracle Linux guests
- Overview of Installing Oracle products
- Installing Oracle Database 10.2.0.3 on SLES10

This chapter is not meant to cover the detailed installation process for Oracle products such as the installation of a 10.2.0.2 database, which is described in Chapter 3, “Installing an Oracle 10gR2 single instance” on page 19.

Remember to follow the latest Oracle documentation and MetaLink notes.

2.1 Setting up VM and Linux guests for Oracle

Here we discuss setting up VM and Linux guests for Oracle.

z/VM

IBM is continually enhancing z/VM to improve the performance of the solution, such as Oracle database and applications. Try to use the latest releases of VM for the best results. Oracle development is presently using VM 5.2 and VM 5.3. You should be at least at these levels.

It is important to make sure that VM has allocated sufficient resource to your Oracle Linux guest. Oracle may not be able to run in your typical Linux guest depending on your default options. For example, you will need:

- Memory

  The Oracle Universal Installer checks to see that you have 1 GB of memory to run the installation of the Oracle code.

  When running the database, memory is needed for Linux, the SGA, and the PGA for the database. If you are using ASM, you need memory for the SGA of the ASM infrastructure database.

  Less is better with Linux on VM so that VM is not busy managing memory that is not necessary.

- Disk

  To install your first Oracle database, you will need disk space for:

  - A copy of the CD images from otn.oracle.com
  - A copy of any patches or patch sets from metalink.oracle.com
  - Your ORACLE_HOME for the Oracle installed code
  - Your Oracle database files and redo logs

  The DBF files should be on a separate mount point from the redo logs for a production database. For subsequent installations, you can NFS mount the CD images and patches to save disk space.

  We usually started with a 20 GB logical disk for our installations of the database. For Oracle applications such as EBS or PeopleSoft, we usually start with a 100 GB LUN. Your disk sizes might differ significantly from what we used.

Note: On IBM System z, you can use LVM with Oracle Single Instance if you want to create one large logical volume from several small ECKD™ devices. For example, if you have two 7 GB drives and you need an 8 GB drive to install Oracle, you can use LVM to create a larger logical volume.

- CPUs

  For testing, a Linux guest with one CPU allocated will be fine. For production, having at least two CPUs will allow for multi-tasking or multi-processing.

- Swap space

  We use 500 MB of swap space. This is sufficient for Oracle on System z Linux if you are running under z/VM. Be aware that the Oracle installation document recommends 1.5 times the memory allocated to Linux. That is an x86 statement and is not recommended here. When the Oracle installer (OUI) runs, it checks for the swap space installed and will flag this as a problem. If this is the only parameter that it flags, then continue the installation. Any other parameters flagged must be fixed. Paging and swapping space are necessary.

Linux

Linux distributors are continually producing new versions (distros). Make sure the one you are using is certified with the Oracle product you are installing. You can check this certification at http://www.metalink.oracle.com. In some cases, you will need a patch to use an Oracle product on the latest distro. This was the case for SLES10 and the Oracle Database 10.2.0.3, as described in 2.3.1, “Installing an Oracle database on SLES10” on page 16.

Oracle requires more RPMs than in a minimum install of Linux. It can be a very long frustrating experience to start with a minimum install and add more RPMs as problems are encountered. The solution of doing an installation of all RPMs, which works well, takes a great deal of disk space and gives you a large Linux guest with RPMs that are never used. Oracle has recently produced several MetaLink notes that can give you guidance in the area of which RPMs to install. You should reference these for the latest information. We list in this section the current information we have for Red Hat 4, SLES9, and SLES10.

Oracle’s recommendations for Linux RPMs

In 2007, Oracle published MetaLink notes to help identify the RPMs needed for each Linux distro level for each Oracle Database product. Metalink note 401705.1 Linux x86 and Linux x86-64 Requirements Reference List is a master list of all the other notes. We have indicated in each of the following sections the

Doc ID for the ones for IBM System z. You should refer to these for the latest information.

Red Hat 4 RPMs

We installed Red Hat AS4 U3, which was the current patch set at the time of the documentation of our install. We installed Red Hat per the IBM recommended process detailed at:

http://www-128.ibm.com/developerworks/eserver/library/es-rhel-coexist/

It is very important that when Red Hat is installed as stated above that the following selections are used:

- Legacy and Compatibility ARCH Development Support
- Legacy Software Development
- Compatibility ARCH Development Support

Without these selections, you will have difficulty installing Oracle. The OUI may not start, or you may see Java™ errors or make/link errors. These will be very time consuming to diagnose.

In addition to the selections listed above, the other functions that we recommend installing are:

- Editors
- DNS
- FTP

Important: The procedure we use to install Novell's SuSE SLES9 for Oracle is listed in 3.6, "Summary" on page 59.

After the installation, we ran the **unname -a** command to verify the results:

```
[oracle@linux23 bin]$ uname -a
Linux linux23.itso.ibm.com 2.6.9-34.EL #1 SMP Fri Feb 24 16:45:04 EST
2006
s390x s390x s390x GNU/Linux
```

The next step we took was to verify that the following RPMs required by Oracle were installed during the installation of Red Hat. The ones that were still missing we installed through the **rpm** command.

The following lines list the RPMs on our Red Hat 4 U3 guest:

```
binutils-2.15.92.0.2-15
compat-db-4.1.25-9
compat-libstdc++-33-3.2.3-47.3
control-center-2.8.0-12.rhel4.2
gcc-3.4.4-2
gcc-c++-3.4.4-2
glibc-2.3.4-2.13
glibc-common-2.3.4-2.13
glibc-devel-2.3.4-2.13
glibc-headers-2.3.4-2.13
glibc-kernheaders-2.4-9.1.98.EL
gnome-libs-1.4.1.2.90-44.1
libaio-0.3.103-3
libaio-devel-0.3.103-3
libgcc-3.4.4-2
compat-libgcc-295-2.95.3-81
libstdc++-3.4.4-2
libstdc++-devel-3.4.4-2
make-3.80-5
ORBit-0.5.17-14
pdksh-5.2.14-30.3
sysstat-5.0.5-1
xorg-x11-deprecated-libs-6.8.2-1.EL.13.20
xscreensaver-4.18-5.rhel4.9
```

See also Oracle's *MetaLink Note 420382.1 Requirements for Installing Oracle 10gR2 RDBMS on RHEL 4 on zLinux (s390x).*

SLES9 RPMs

See *MetaLink Note Requirements for Installing Oracle 10gR2 RDBMS on SLES 9 zLinux (s390x) 431443.1* for Oracle's recommendations for the RPMs you need for Oracle 10gR2.

For Oracle 9i, see Oracle *MetaLink Note 270577.1 Installing Oracle 9i on IBM z/Series - SLES8/9.*

SLES10 RPMs

See Oracle *MetaLink Note:415182.1 DB Install Requirements Quick Reference - zSeries based Linux.*

When you install 10.2.0.2 on SLES10, you will encounter an error that is described in *Metalink Note 417001.1 Errors installing 10.2.0.2 patchset on IBM ZSeries Based Linux.*

Oracle 10.2.0.2 is not certified on SLES10. You must install a patch to bypass this error and then install the 10.2.0.3 patch set. This is described in detail in 2.3.1, “Installing an Oracle database on SLES10” on page 16.

We used the **rpm -qa** command to display our RPMs:

```
gcc-3.3.3-43.34
gcc-c++-3.3.3-43.34
glibc-2.3.3-98.47
libgcc-3.3.3-43.34
libstdc++-3.3.3-43.34
libstdc++-devel-3.3.3-43.34
libaio-0.3.102-1.2
libaio-devel-0.3.102-1.2
make-3.80-184.1
glibc-devel-32bit-9-200506070135
```

Check Linux ulimit parameters

Run:

```
ulimit -n 65536
ulimit -u 16384
ulimit -p 16384
```

Check Linux Kernel parameters

Run:

```
SEMMSL 250
SEMMNS 32000
SEMOPM 100
SEMMNI 128
SHMMAX 2147483648 or half the size of your system's physical memory in
bytes
SHMMNI 4096
SHMALL
FILE-MAX 65536
IP_LOCAL_PORT_RANGE 1024-65000
RMEM_DEFAULT 1048576
RMEM_MAX 1048576
WMEM_DEFAULT 262144
WMEM_MAX 262144
```

Update the Oracle user ID shell limits

Run:

```
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
```

Update the pam limits

Run:

```
session required /lib/security/pam_limits.so
session required pam_limits.so
```

Check permissions for the mount point that the oracle user ID must access:

```
Oracle staging area
Oracle patches
ORACLE_HOME for the binaries
DBF files
Redo log files
```

2.2 Installing Oracle products

This section lists where we have described our installation experiences with several of the Oracle products.

2.2.1 Installing an Oracle database

Our experiences with installing a single instance database are described in detail in Chapter 3, “Installing an Oracle 10gR2 single instance” on page 19. For information about using the Real Application Clusters (RAC) feature, see Chapter 4, “Setting up Oracle RAC on Linux for System z” on page 63.

2.2.2 Installing Oracle AS10g

Our experiences with installing the components of AS10g are described in *Experiences with Oracle?® 10gR2 Solutions on Linux for IBM System z*, SG24-7191.

2.2.3 Installing Oracle applications

Check the latest information in the Oracle documentation and on MetaLink. Our experiences with EBS 11i are described in *Experiences with Oracle?® 10gR2 Solutions on Linux for IBM System z*, SG24-7191.

2.3 Installing patches for 10gR2 on SLES10

Because of some changes to the Linux distributions, there are special patches that must be installed in some cases. We describe these patches in detail for SLES10 in 2.3.1, “Installing an Oracle database on SLES10” on page 16.

2.3.1 Installing an Oracle database on SLES10

As of June 2007, Oracle has certified 10.2.0.3 on SLES10. Installing a single instance database on SLES10 requires some additional steps.

The steps for a new installation are as follows:

1. Install SLES10. We recommend that you do an “install all” to get all the components of SLES10 for your first test environment. If you choose to only install components, make sure that you include all the gcc components and compat libraries for 31- and 64-bit.
2. Obtain the Oracle code and documentation.
   a. Download Oracle 10.2.0.2 for Linux for System z from otn.oracle.com.
   b. Download the 10.2.0.2 Release Notes and Installation Guide.
   c. Use the following IBM Redbooks as guides: *Experiences with Oracle?® 10gR2 Solutions on Linux for IBM System z*, SG24-7191 and this one.
   d. Download patch 6007358 from http://www.metalink.oracle.com so that 10.2.0.2 can install on SLES10. This patch is only found with an IBM S390 Linux platform with a June 2007 date; it should be moved to the IBM zSeries® Linux platform.
   e. Download patch set 5337014 for 10.2.0.3 and the patch set notes from http://www.metalink.oracle.com. This patch set is found with the Linux for zSeries with a June 2007 date.

Note: The Patch set README file refers to the incorrect file name in section 5.4 “Download and Extract the Installation Software”. The correct name is p5337014_10203_LINUX-zSer.zip, not p5337014_10203_LINUX-S390.zip patch set. The same README file refers to the incorrect patch name in section 10.13 - “10.2.0.2 Install on SUSE Linux Enterprise Server 10 (SLES10) fails Installing 10.2.0.2 on SUSE Linux Enterprise Server (SLES) 10 fails”. The workaround is to install the 6017970 patch, which should read “install the 6007358 patch”.

3. Run the OUI for 10.2.0.2 on SLES10 until you get the link error. Do not exit. Open a separate window to install the patch. Just leave the OUI running while you install patch 6007358 from another window. After installing the patch, return to the OUI and click **Retry**.

The manual steps to follow to apply the one-off patch are:

1. Unzip the patch zip file into a temp directory, for example, to /oracle1/shiphomes/p6007358. This creates an appropriate patch directory structure that has all the necessary files in it.
2. Copy the libraries to the appropriate directory under $ORACLE_HOME. For example:

```
cp /oracle1/shiphomes/p6007358/6007358/files/lib/libclient10.a
ORACLE_HOME/lib/libclient10.a
          cp
/oracle1/shiphomes/p6007358/6007358/files/lib32/libclient10.a
$ORACLE_HOME/lib32/libclient10.a
```

3. Click **Retry** on the error message box in the OUI.

This allows the 10.2.0.2 installation on SLES10 to complete, and then you can upgrade to 10.2.0.3 with patchset 5337014.

Note: This same patch p6007358 is needed to install the Oracle Client on SLES10.

3

Installing an Oracle 10gR2 single instance

In this chapter, we have included the installation of a single instance database 10gR2 for Linux for System z. This is an updated version of the chapter of our previous book, *Experiences with Oracle?® 10gR2 Solutions on Linux for IBM System z*, SG24-7191.

We followed this procedure several times in the course of writing the chapters in this IBM Redbooks publication.

This chapter includes the following topics:

- Preparing the environment:
  - z/VM environment
  - Linux environment
  - Linux kernel parameters
  - Setting up X Window System
- Preparing to install
- Installation
- Post installation

3.1 Installing Oracle Database 10gR2 on Linux for System z

Oracle Database 10gR2 for Linux for System z is 64-bit. You can set it up using a Red Hat AS4 Update 3 or Novell SUSE Linux Enterprise Server (SLES9) SP3 64-bit Linux or SLES10 guest under z/VM in a logical partition (LPAR) on System z or in an LPAR running only Linux.

The version released in May 2006 was Oracle Database 10gR2 (10.2.0.2). We based this document on the installation experiences that we gained when installing this code at IBM and Oracle locations.

This chapter assumes that you have a functional Linux image. It describes the steps that we performed to prepare the environment and to run the installation process.

3.2 Preparing the environment for the installation

We based this chapter on a Linux virtual machine named *linux23*. We installed Red Hat AS4 U3. We used a file system that was a 7 GB ext3 Logical Volume Manager (LVM) mounted as /oradbf. We used a z/VM LPAR and z/VM V5.2. We chose to use an LVM instead of Oracle Automatic Storage Management (ASM). Either one is a valid option.

3.2.1 z/VM V5.2 definitions for the Linux virtual machine

We defined a Linux virtual machine with 2 GB of virtual memory and one virtual CPU. Example 3-1 shows the user directory entries for our virtual machine, linux23.

Example 3-1 linux23 virtual machine definitions

```
USER LINUX23 xxxxxxxx 2G 2G G
   ACCOUNT ITS30000
   IPL CMS PARM AUTOCR
   MACHINE XA
   NICDEF C200 TYPE QDIO LAN SYSTEM VSWITCH1
   CONSOLE 0009 3215
   SPOOL 000C 3505 A
   SPOOL 000D 3525 A
   SPOOL 000E 1403 A
   LINK MAINT 0190 0190 RR
```

```
LINK MAINT 019E 019E RR
LINK MAINT 019F 019F RR
LINK MAINT 019D 019D RR
MDISK 0191 3390 2062 50 VMLU1A MR
MDISK 0201 3390 0001 0200 LX1A3C MR
MDISK 0202 3390 0201 3038 LX1A3C MR
MDISK 0203 3390 0001 3338 LX1B40 MR
MDISK 0204 3390 0001 3338 LX1B47 MR
MDISK 0301 3390 0001 10016 LXC40B MR
MDISK 0302 3390 0001 10016 LXC40C MR
```

We added 500 MB of swap space. This is sufficient for Oracle on Linux for System z. Be aware that the Oracle installation document recommends 1.5 times the memory allocated to Linux. That is an x86 statement and we do not recommend it here. When the Oracle installer (the Oracle User Interface (OUI)) runs, it checks for installed swap space and flags this as a problem. If this is the only parameter that it flags, then you can continue the installation. You must fix any other parameters that are flagged.

3.2.2 The Linux environment

The system administrator needs to set up the Linux guest with resources such as CPU, memory, disk, and network connectivity that are specific to the installation's needs.

We installed Red Hat AS4 Update 3, which was the current patch set at the time of writing. We installed Red Hat using the IBM recommended process found at:

http://www-128.ibm.com/developerworks/eserver/library/es-rhel-coexist/

It is very important that when you install Red Hat that you use the following selections:

- Legacy and Compatibility ARCH Development Support
- Legacy Software Development
- Compatibility ARCH Development Support

Without these settings, it is difficult to install Oracle. The Oracle User Interface (OUI) might not start, and you might see Java errors or make/link errors. These situations are time consuming to diagnose.

We recommend that you install these functions, as well as the functions listed previously:

- Editors
- DNS
- FTP

Important: We list the procedure that we use to install SLES9 for Oracle in 3.6, “Summary” on page 59.

After the installation, we ran the **unname -a** command to verify the results:

```
[oracle@linux23 bin]$ uname -a
Linux linux23.itso.ibm.com 2.6.9-34.EL #1 SMP Fri Feb 24 16:45:04 EST
2006 s390x s390x s390x GNU/Linux
```

Next, we verified that we installed the following Red Hat Package Managers (RPMs) that Oracle requires during the installation of Red Hat. We installed the RPMs that were still missing through the **rpm** command.

Example 3-2 List of RPMs on our Red Hat AS4 Update 3 guest

```
binutils-2.15.92.0.2-15
compat-db-4.1.25-9
compat-libstdc++-33-3.2.3-47.3
control-center-2.8.0-12.rhel4.2
gcc-3.4.4-2
gcc-c++-3.4.4-2
glibc-2.3.4-2.13
glibc-common-2.3.4-2.13
glibc-devel-2.3.4-2.13
glibc-headers-2.3.4-2.13
glibc-kernheaders-2.4-9.1.98.EL
gnome-libs-1.4.1.2.90-44.1
libaio-0.3.103-3
libaio-devel-0.3.103-3
libgcc-3.4.4-2
compat-libgcc-295-2.95.3-81
libstdc++-3.4.4-2
libstdc++-devel-3.4.4-2
make-3.80-5
ORBit-0.5.17-14
pdksh-5.2.14-30.3
sysstat-5.0.5-1
xorg-x11-deprecated-libs-6.8.2-1.EL.13.20
xscreensaver-4.18-5.rhel4.9
```

Setting up the group and user ID for oracle

The steps are:

1. We log in as root.
2. We create two groups, dba and oinstall, by running the **groupadd** command:

   ```
   groupadd oinstall
   groupadd dba
   ```

3. We create the user oracle with this command:

   ```
   useradd -g oinstall -G dba oracle
   ```

4. We complete this sequence of tasks by creating a password for oracle by using the **passwd** command.
5. We use the list command, **ls -la,** in the / directory to see who owns the mount point /oradbf:

   ```
   [root@linux23 /]# ls -la
   total 188
   drwxr-xr-x  26 root   root  4096 Oct  1 12:58 .
   drwxr-xr-x  26 root   root  4096 Oct  1 12:58 ..
   -rw-r--r--   1 root   root     0 Sep 13 06:58 .autofsck
   drwxr-xr-x   2 root   root  4096 Jul 13 11:52 bin
   drwxr-xr-x   3 root   root  4096 Jul 13 12:03 boot
   drwxr-xr-x   9 root   root  3060 Sep 13 06:58 dev
   drwxr-xr-x  71 root   root  8192 Oct  2 04:02 etc
   drwxr-xr-x   4 root   root  4096 Sep 14 17:14 home
   drwxr-xr-x   2 root   root  4096 Aug 12  2004 initrd
   drwxr-xr-x   8 root   root  4096 Jul 13 13:39 lib
   drwxr-xr-x   6 root   root  4096 Jul 13 11:57 lib64
   drwx------   2 root   root 16384 Jul 13 11:48 lost+found
   drwxr-xr-x   2 root   root  4096 Aug 12  2004 media
   drwxr-xr-x   2 root   root  4096 Dec  2  2005 misc
   drwxr-xr-x   2 root   root  4096 Aug 12  2004 mnt
   drwxr-xr-x   4 root   root  4096 Aug  8 10:03 opt
   drwxr-xr-x   4 oracle dba   4096 Oct  1 15:30 oracle
   drwxr-xr-x  10 root root 4096 Oct  1 13:01 oradbf
   dr-xr-xr-x  72 root   root     0 Sep 13 06:58 proc
   drwxr-x---   6 root   root  4096 Oct  5 09:52 root
   drwxr-xr-x   2 root   root  4096 Jul 13 11:53 sbin
   drwxr-xr-x   2 root   root  4096 Jul 13 11:49 selinux
   drwxr-xr-x   2 root   root  4096 Aug 12  2004 srv
   drwxr-xr-x   8 root   root     0 Sep 13 06:58 sys
   drwxrwxrwt   4 root   root  4096 Oct  5 09:49 tmp
   drwxr-xr-x  16 root   root  4096 Jul 13 11:51 usr
   drwxr-xr-x  21 root   root  4096 Jul 13 12:00 var
   ```

```
[root@linux23 /]#
```

6. Ensure that the user oracle can write to the /oracle directory. We created the mount point as part of the installation as root. Because our directory was created by the user ID root, we had to issue the following command:

```
chown oracle:dba oracle
chmod 755 /oracle
```

7. Reissue the **ls -la** command at the / directory and verify that the ownership is correct. The /oradbf directory should be:

```
drwxr-xr-x  10 oracle dba   4096 Oct  1 13:01 oradbf
```

8. We also verified our name resolution on our Linux virtual machine according to the *Oracle Database Installation Guide 10g Release 2 (10.2) for IBM System z Based Linux,* B25400-01, May 2006 (new book in 10gR2). It is important the /etc/hosts file has the IP address of this virtual machine, the fully qualified domain name, and the short name. Our /etc/hosts file had the following entry:

```
9.12.4.173 linux23.itso.ibm.com linux23
```

3.2.3 Linux kernel parameters

Next, we check the kernel parameters and set the limits for open files and processes for the oracle user. The steps are:

1. First, we make the entries in Example 3-3 in /etc/sysctl.conf.

Example 3-3 sysctl.conf entries

```
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1
```

```
#Kernel parameter changes for Oracle 10g R2
kernel.shmall=2097152
kernel.shmmax=2147483648
kernel.shmmni=4096
kernel.sem=250 32000 100 128
fs.file-max=65536
net.ipv4.ip_local_port_range=1024 65000
net.core.rmem_default=1048576
net.core.rmem_max=1048576
net.core.wmem_default=262144
net.core.wmem_max=262144
```

2. After making these changes, we reboot Linux to implement these changes.
3. Next, we changed the limits for the oracle user, as shown in Example 3-4.

Example 3-4 limits.conf file

```
#<item> can be one of the following:
#        - core - limits the core file size (KB)
#        - data - max data size (KB)
#        - fsize - maximum filesize (KB)
#        - memlock - max locked-in-memory address space (KB)
#        - nofile - max number of open files
#        - rss - max resident set size (KB)
#        - stack - max stack size (KB)
#        - cpu - max CPU time (MIN)
#        - nproc - max number of processes
#        - as - address space limit
#        - maxlogins - max number of logins for this user
#        - maxsyslogins - max number of logins on the system
#        - priority - the priority to run user process with
#        - locks - max number of file locks the user can hold
#        - sigpending - max number of pending signals
#        - msgqueue - max memory used by POSIX message queues (bytes)
#
#<domain>      <type>  <item>         <value>
#

#*               soft    core            0
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
```

```
#ftp             hard    nproc           0
#@student        -       maxlogins       4
oracle  soft    nproc   2047
oracle  hard    nproc   16384
oracle  soft    nofile  1024
oracle  hard    nofile  65536
"/etc/security/limits.conf" 51L, 1799C
```

4. The changes in Example 3-4 on page 25 determine the maximum limits for oracle.
5. Next, we change the /etc/profile file so that the oracle user is able to set these limits at logon time, as shown in Example 3-5.

Example 3-5 Changes to the /etc/profile file for the oracle user

```
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

pathmunge () {
        if ! echo $PATH | /bin/egrep -q "(^|:)$1($|:)" ; then
           if [ "$2" = "after" ] ; then
              PATH=$PATH:$1
           else
              PATH=$1:$PATH
           fi
        fi
}

# Path manipulation
if [ `id -u` = 0 ]; then
        pathmunge /sbin
        pathmunge /usr/sbin
        pathmunge /usr/local/sbin
fi

pathmunge /usr/X11R6/bin after

#defaults for  shell startup for oracle
if [ $USER="oracle" ]; then
        if [ $SHELL "/bin/ksh" ]; then
                ulimit -p 16384
                ulimit -n 65536
        else
```

```
                ulimit -u 16384 -n 65536
        fi
fi

# No core files by default
ulimit -S -c 0 > /dev/null 2>&1

USER="`id -un`"
LOGNAME=$USER
```

6. Then, we make changes to the /etc/pam.d/login file using the installation document so that the oracle user is forced to look at the limits.conf file to allow it to get the limits listed in the file.

   Using the installation document, we added:

   ```
   session    required    pam_limits.so
   ```

   Note: When we added `session required /lib/security/pam_limits.so`, we received error messages from the VMconsole, even though the line is recommended in the Oracle documentation. We removed this line and the errors messages disappeared.

7. We use VNC Server on the Linux guest. This works for that configuration. If you use other communication tools, such as XDM, you might need to add the pam_limits.so parameter to /etc/pam.d/xdm or /etc/pam.d/su. You see this problem when you run **logon** or **su**, because you get a bash message indicating that the user is not authorized to set limits.

We also advise that when you make changes to files in pam.d that you realize that the changes are enabled when you log off and then log in again. If you make mistakes here, then you might never be able to log in again. Therefore, after you make changes as root, we advise that you stay logged on as root and log oracle off and on to make sure everything works, as shown in Example 3-6.

Example 3-6 Settings for /etc/pam.d/login

```
#%PAM-1.0
auth       required     pam_securetty.so
auth       required     pam_stack.so service=system-auth
auth       required     pam_nologin.so
account    required     pam_stack.so service=system-auth
password   required     pam_stack.so service=system-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
```

```
session    required     pam_stack.so service=system-auth
session    required     pam_loginuid.so
session    required     /lib/security/pam_limits.so
session    required    pam_limits.so
session    optional     pam_console.so
# pam_selinux.so open should be the last session rule
session    required     pam_selinux.so open
```

3.2.4 X Window System interface setup using VNC

The Oracle Universal Installer (OUI) requires that you have an X Window System interface. To enable the X Window System interface, we used VNC, which comes with SUSE Linux. We downloaded a VNC viewer from the following Web site:

`http://www.realvnc.com/download.html`

You can download either a Linux or Windows® version, depending on the client that you chose for the installation.

After you have a VNC client installed, do the following steps to set up the viewer:

1. Using PuTTY, log on using the secure shell. We logged in to the Linux guest as oracle. To start the VNC server, we issued the command:

   `vncserver`

2. This starts the VNC server for the oracle user. The system prompts you for a password and to verify the password that you entered. Start the VNC server with **`vncserver`**. It starts with the same session as before and retains your password.

   You can run many VNC sessions with different user IDs for the sessions using the motif viewer for oracle or any other user ID. The .vnc directory and associated files are located in /home/*user ID*/.vnc.

   Note that different users have different ports. In our case, user root was:

   `9.12.4.172:1`

   The oracle user ID was:

   `9.12.4.172:2`

   We use two VNC sessions: one for oracle and one for root. This seems simpler than switching users (**`su`**) within the same session. This is also more likely the environment that a typical DBA experiences.

3. After we start the session, we right-click in the window. This action displays a work menu from which we select the terminal to open a second window for

the user ID oracle. In this situation, we use the **su** command to change to user ID root.

We discuss and demonstrate this in 3.4, “Installation” on page 32. First, we must complete the preparation.

3.3 Preparing for the installation

We must download the code and get the documentation we need for the installation.

3.3.1 Downloading the code

Important: If you download the database code from the Oracle Technical Network (OTN), it does not come with a license. This means that there is no Oracle support associated with this code if you merely download the code. Clients can purchase a license, or if they have a virtual license (an unused license), clients can apply the virtual license to the downloaded version. If a client has any questions or concerns about obtaining a version of the database software and how to handle licenses or support, the client needs to discuss this with their Oracle sales representative.

In the following steps, we download the required software to a Windows computer, and download the zip files to our linux23 machine.

Do these steps to download a copy of the database software:

1. Go to:

   `http://www.oracle.com/technology/index.html`

2. Select **Downloads** → **Database**.
3. Select **Oracle Database 10gR2 for Linux for System z**.
4. Download the file 10202_zlinux_database.zip.
5. After you download the code, go to the directory on the Linux guest where you want to store the images and complete the following commands:

   ```
   cd /oradbf
   mkdir images
   cd images
   ```

   Then, FTP to the Linux guest and place the images in the images directory.

We received the database and the Companion CD files:

```
10202_zlinux_database.zip for the database
10202_zlinux_companion.zip for the companion products
```

Use the **ls -la** command to check that you received the same size files as on the host.

6. Uncompress the database file in the /oracle/images directory:

```
unzip 10202_zlinux_database.zip
```

These are large files. The **ftp** and **unzip** commands can take several minutes to complete.

3.3.2 Finding the documentation

You can obtain documentation for this code at:

http://www.oracle.com/technology/index.html

Click **Documentation** at the top of the page and then click **Database**.

Make sure that you obtain:

- The release notes for Oracle Database 10gR2: *Oracle Database Release Notes 10g Release 2(10.2) for IBM zSeries Based Linux,* B25399
- *Oracle Database Installation Guide for IBM zSeries Based Linux*, B25400

The release notes contain important information that is not included in the Installation guides.

Follow the steps in the Oracle documentation for the installation process. This chapter is not meant to replace the Oracle documentation, but we documented our experiences to help guide you through the installation.

3.3.3 Determine naming conventions

Choose the names for ORACLE_HOME, ORACLE_SID, and so on, as shown in Table 3-1.

Table 3-1 Names chosen for Oracle installation

Variable	Value	Comment
ORACLE_BASE	/oradbf	Highest level directory for this Oracle installation.
ORACLE_SID	lnx23orl	Database name.
ORACLE_HOME	/oradbf/product/10gR2	Where the binaries are located.
ORACLE_INVENTORY	/oradbf/oraInventory	Must not be in ORACLE_HOME but under ORACLE_BASE.
Images directory	/oradbf/images/	Where we put the “copy in and copy out” (cpio) files.
Oracle db files	/oradbf/oradata	N/A.

We chose the directories in Table 3-1.

To facilitate the installation, we put the following entries in the oracle users .profile file:

```
export ORACLE_BASE=/oradbf
export ORACLE_HOME=/oradbf/product/10gR2
export ORACLE_INVENTORY=/oradbf/oraInventory
export PATH=$PATH:$ORACLE_HOME/bin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/rdbms/bin
export ORACLE_SID=lnx23orl
```

And then we executed the profile by entering:

```
. .profile
```

(This is a dot followed by a space, then .profile.)

The profile does not need to be set before the installation is complete. In fact, according to the Oracle Installation Guide, only $ORACLE_BASE needs to be set. We did this for convenience, because this is the only Oracle software installed on this Linux installation. If you follow the Installation Guide, then you need to make these additions at the end of the installation.

3.4 Installation

The oracle user must start the OUI:

1. With the VNC server started while logged in as the oracle user, we started the VNC viewer by inputting 9.12.4.172:2, as shown in Figure 3-1, and clicked **OK**.

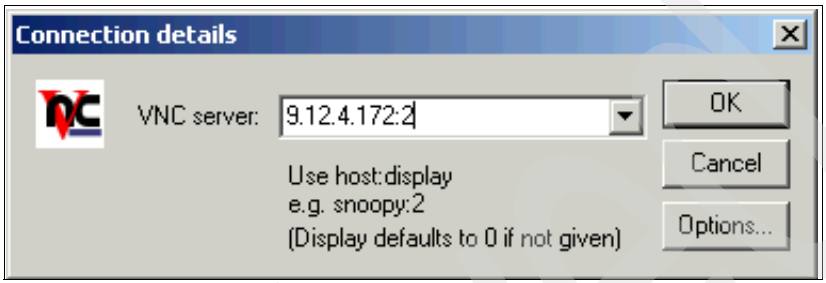


Figure 3-1 VNC server IP address

2. Then, we entered our session password (Figure 3-2) to start the VNC session, and clicked **OK**.

Figure 3-2 VNC password

3. If the **vncserver** command is not started by oracle, then you must telnet (we used PuTTY) into the user ID oracle and start it using the procedure that we described in 3.2.4, “X Window System interface setup using VNC” on page 28.
4. We also ensured that the directories listed in Table 3-1 on page 31 have the ownership of oracle. If you try this from user ID root, you get the following message:

```
The user is root. Oracle Universal Installer cannot continue
installation if the user is root.
: No such file or directory
```

5. We went to the directory where the CD-ROM images were stored, then we ran the installer as the oracle user:

```
cd /images/database
./runInstaller
```

6. This takes a few minutes. The Welcome window appears followed by the window shown in Figure 3-4 on page 34.

Note: You should not have a problem starting the OUI if you have ran the **vncserver** command as oracle. However, if you run the **vncserver** command as root, you have to set the DISPLAY command in your user ID oracle session and issue the **xhost +** command as D root.

If you log on as a user that is not oracle, start the X server (using VNC, Cygwin, or any other method), and then use the **su -l oracle** command; you need to execute the **xhost +** command to allow access to the X server from the Oracle session, so that the display appears. The X server is owned by the user that started it and not by oracle.

Use the **xhost** program to add and delete host names or user names to the list allowed to make connections to the X server. There are several notes in MetaLink that further explain how to do this.

The Oracle Universal Installer (OUI) presents a series of panels that enables you to choose the appropriate options and to enter the information required for the installation process.

If the **vncserver** was not run in oracle, then you must telnet (we use PuTTY) into the user ID oracle and start it by using the procedure described in 3.2.4, “X Window System interface setup using VNC” on page 28**.**

The steps that we performed are:

1. We went to the directory where the CD-ROM images were stored, and then we ran the installer as the oracle user:

```
cd /oradbf/images/database
./runInstaller
```

2. The OUI presents a series of windows for you to choose the appropriate options and to enter the information that is required for you to install the Oracle software and create a database. See Figure 3-3.

Figure 3-3 Starting the OUI

After the Welcome window appears, the next window asks for the installation type. We selected **Advanced Installation**, as shown in Figure 3-4.

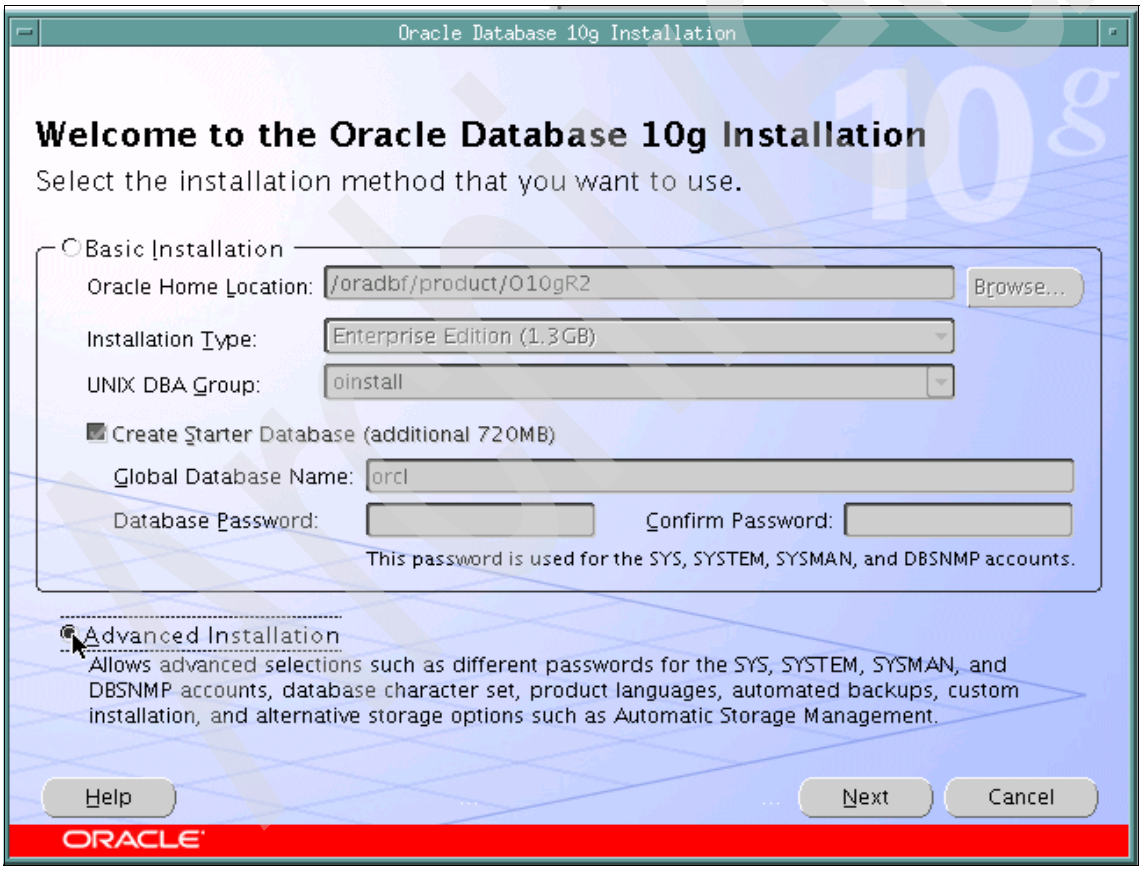


Figure 3-4 Select Advanced Installation

3. We selected the Oracle Database 10g **Enterprise Edition**, as shown in Figure 3-5. The other selections allow the installation of the Standard Edition (SE). This is not allowed in System z, because this version is limited to systems with a physical maximum of four CPUs.

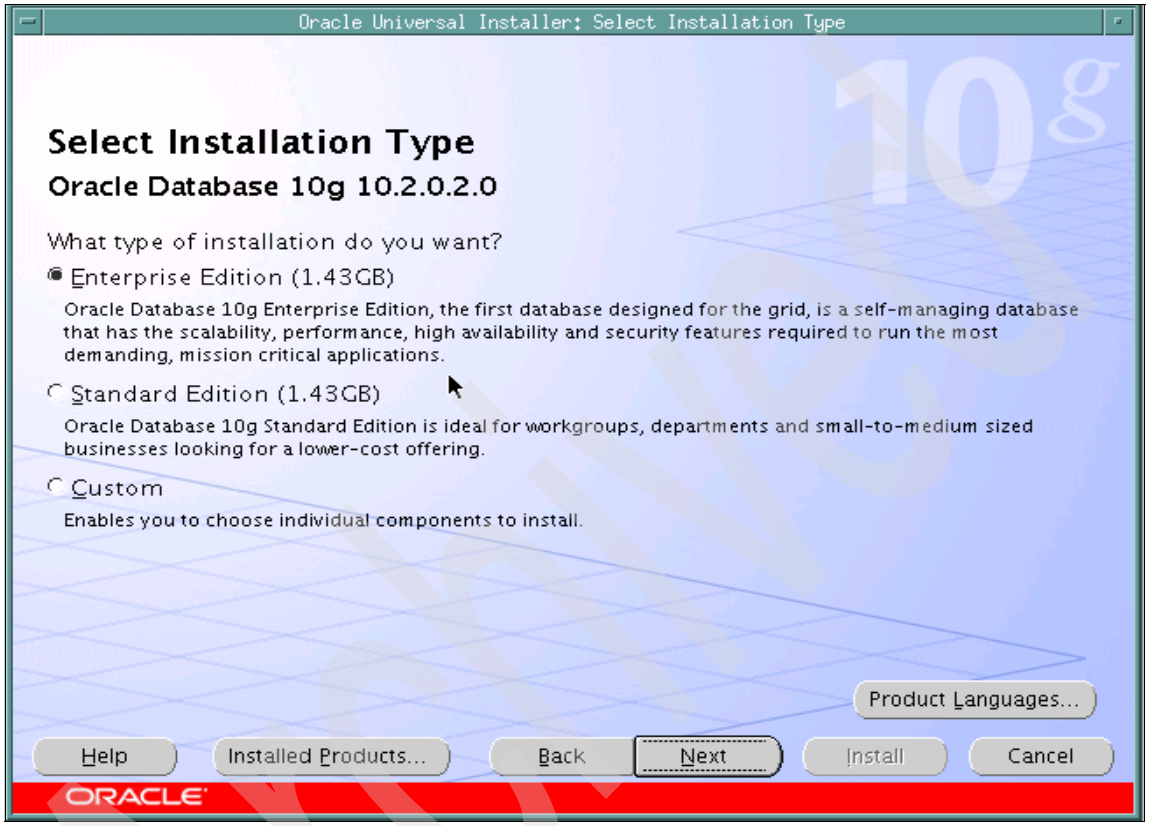


Figure 3-5 Select Installation Type: Enterprise Edition

4. You must indicate your Oracle home directory (Figure 3-6) where you install the Oracle executable software. This location was predetermined by the setting that we made in the oracle profile. The OUI recognized this and placed this value in the Path field for us.

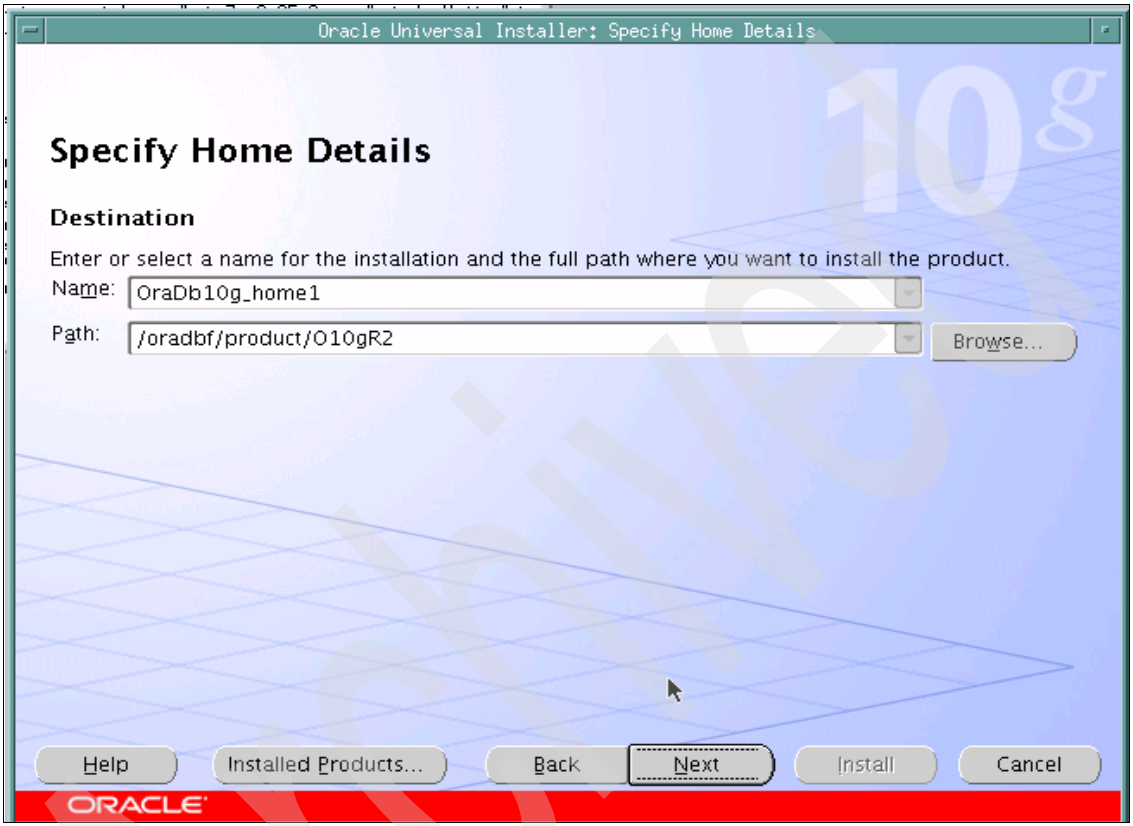


Figure 3-6 Inventory path for first installation on this Linux guest

5. Figure 3-7 on page 37 checks your system against the system prerequisites to ensure that your system is ready for the installation of Oracle binaries. As we stated previously, 512 MB for swap space is sufficient. The OUI uses a formula based on the Intel® platform to determine the proper amount of swap

space. As long as you have installed 512 MB of swap space, ignore the message about insufficient swap space and continue. This section of the installation verifies that other kernel parameters are correct, and if not, warns you that you must correct them.

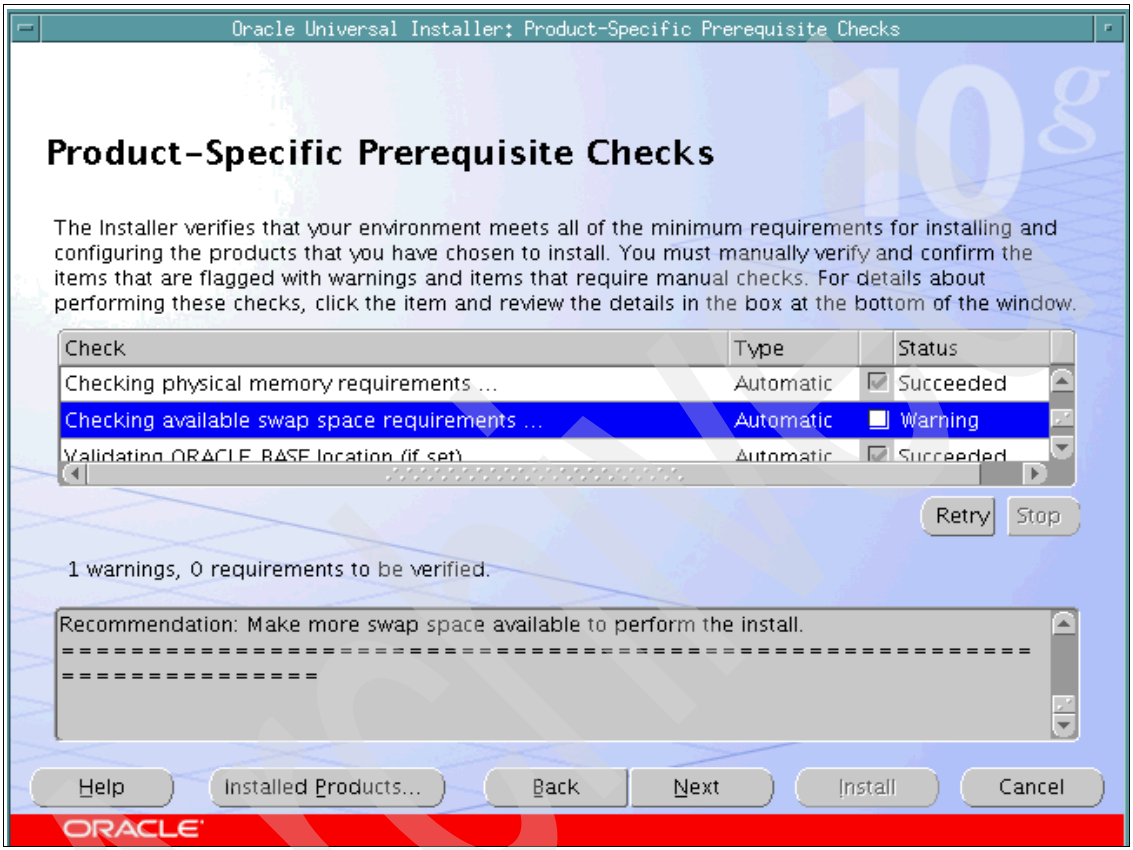


Figure 3-7 Checking prerequisites

6. Figure 3-8 demonstrates that the prerequisite check has failed for insufficient swap space. Again, as long as you have installed 512 MB of swap space, you can ignore this message and continue.

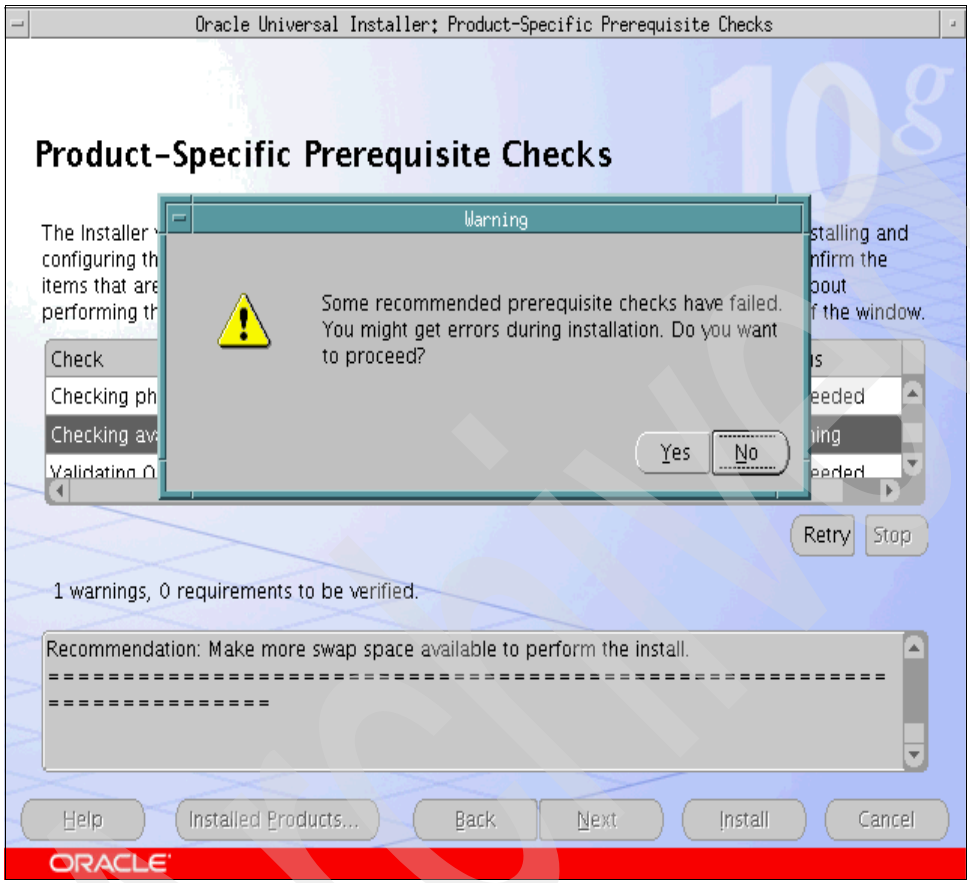


Figure 3-8 Failure with a prerequisite check

7. We selected **Yes** and continued with the installation.

8. The selections in Figure 3-9 offer you the choices of creating and installing a database, configuring Automatic Storage Management (ASM) to manage storage, or just installing the Oracle database software. We selected **Create a database**.

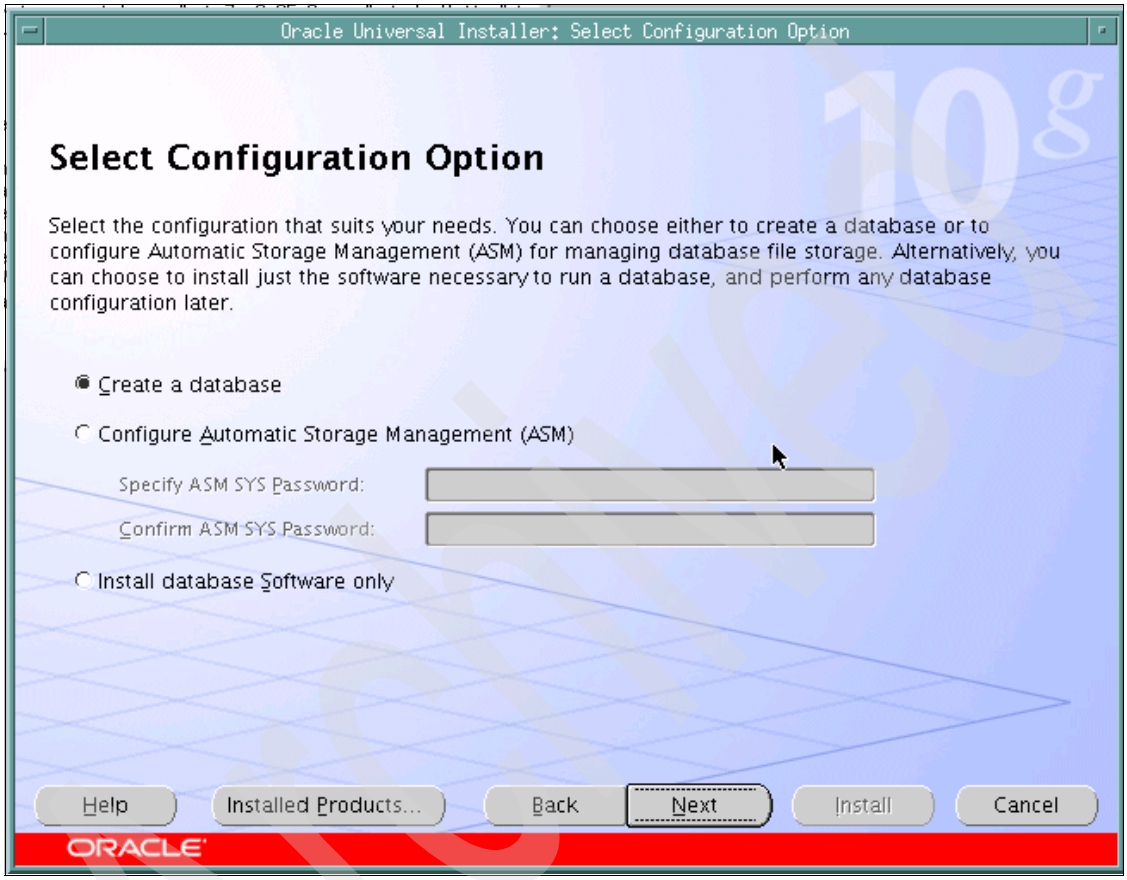


Figure 3-9 Database configuration

9. Figure 3-10 shows the installation of a database for a specific purpose. We selected **General Purpose**.

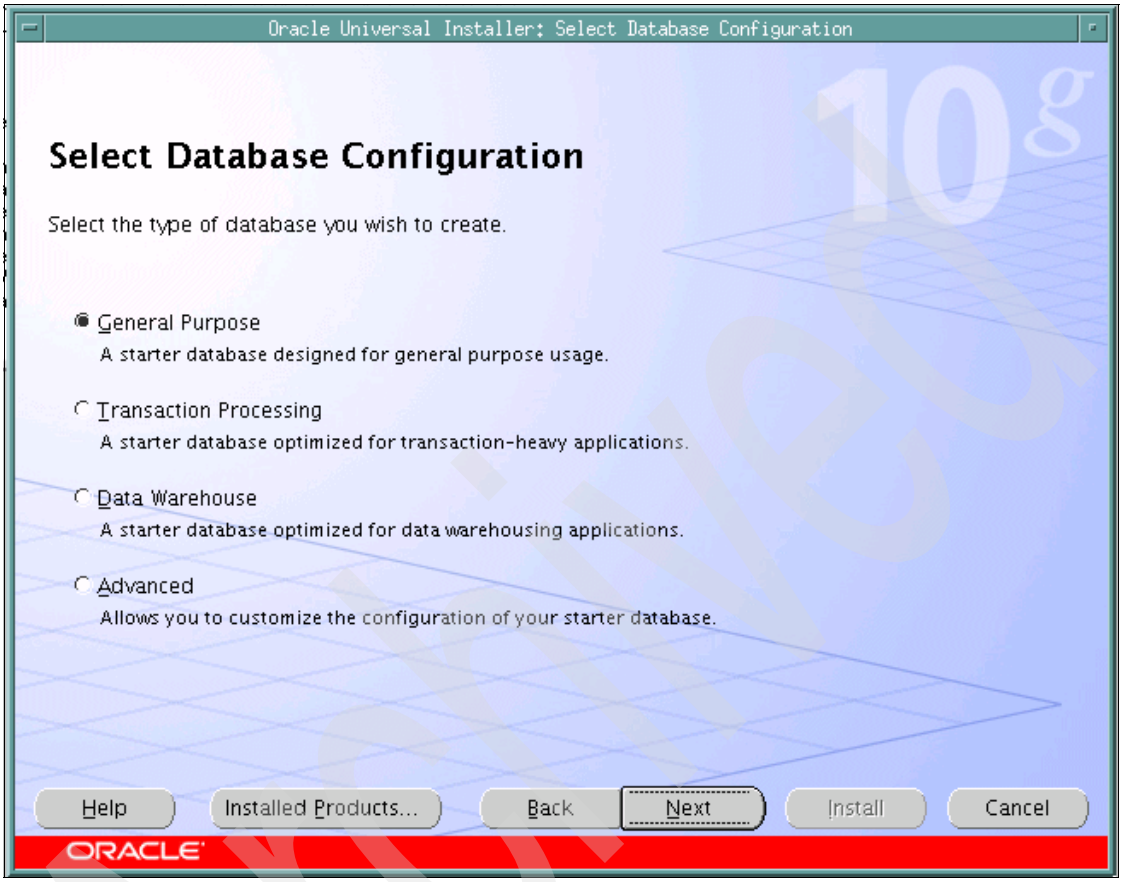


Figure 3-10 Selecting the database configuration

10. We named our database lnx23orl, as shown in Figure 3-11, and accepted the default character set.

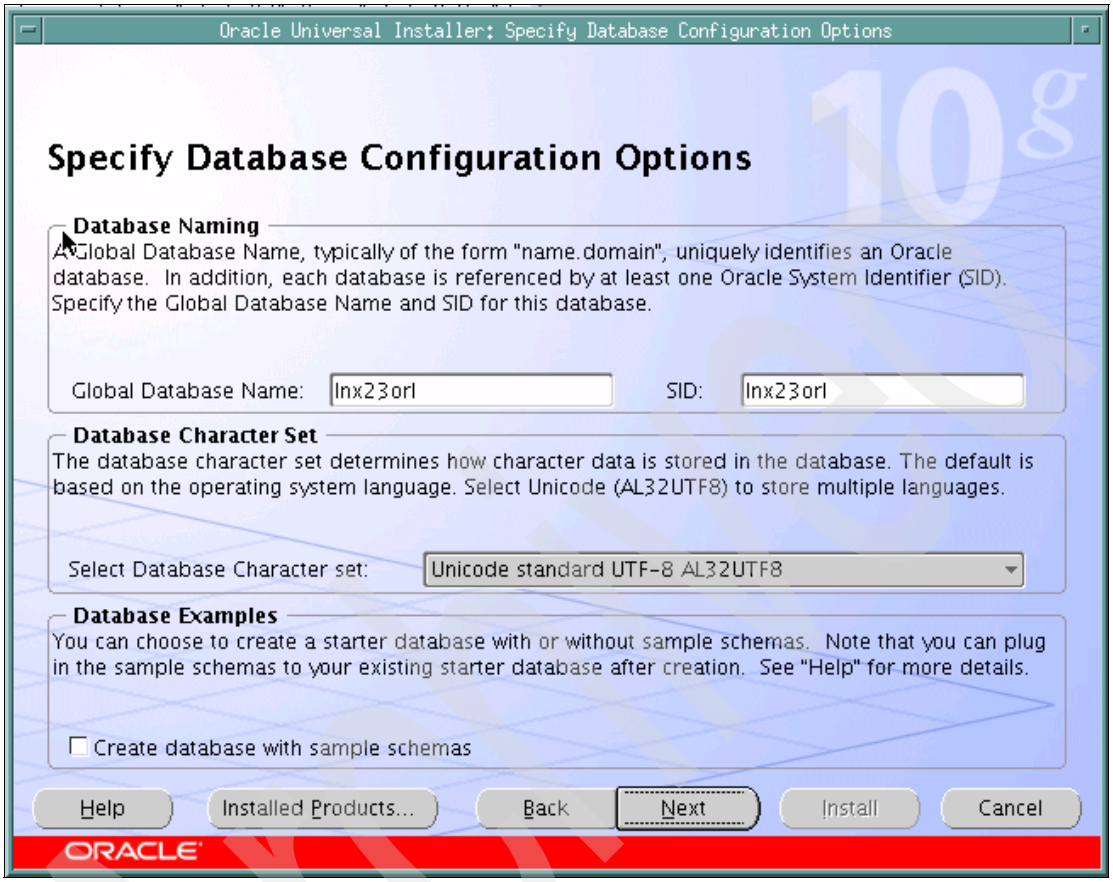


Figure 3-11 Configuration options

11. We selected **Use Database Control for Database Management** for our database management option, as shown in Figure 3-12. The agent can be installed separately from this installation if you want to manage this database from Grid Control. The Grid Control server must be installed on another platform.

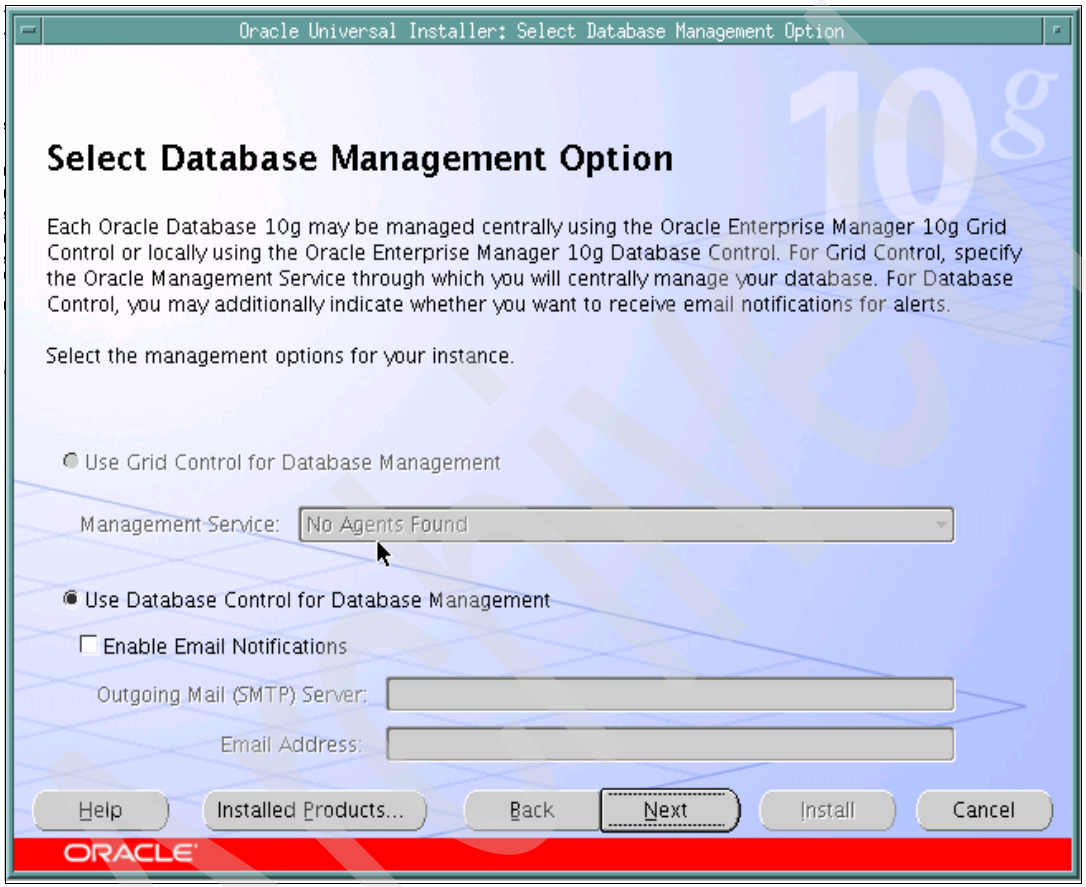


Figure 3-12 Database management options

12. Figure 3-13 shows our selection of a file system and the mount point that we entered.

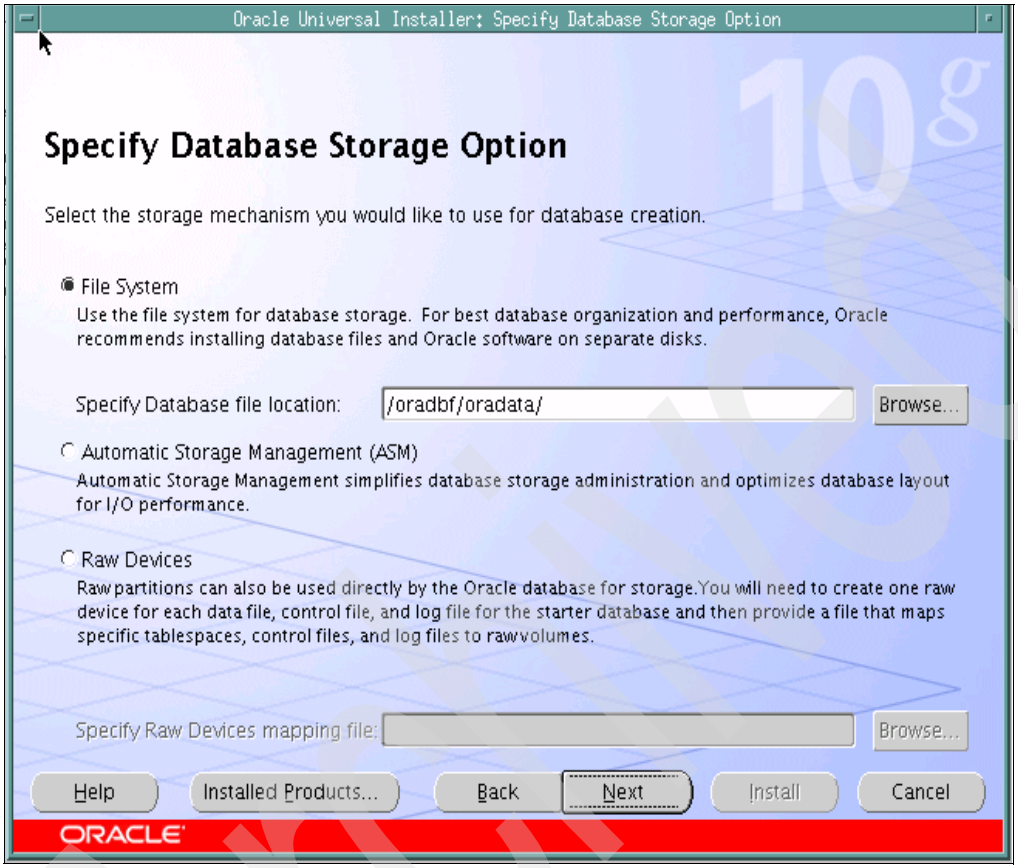


Figure 3-13 Storage option

13.We chose not to enable Automated Backups, as shown in Figure 3-14.

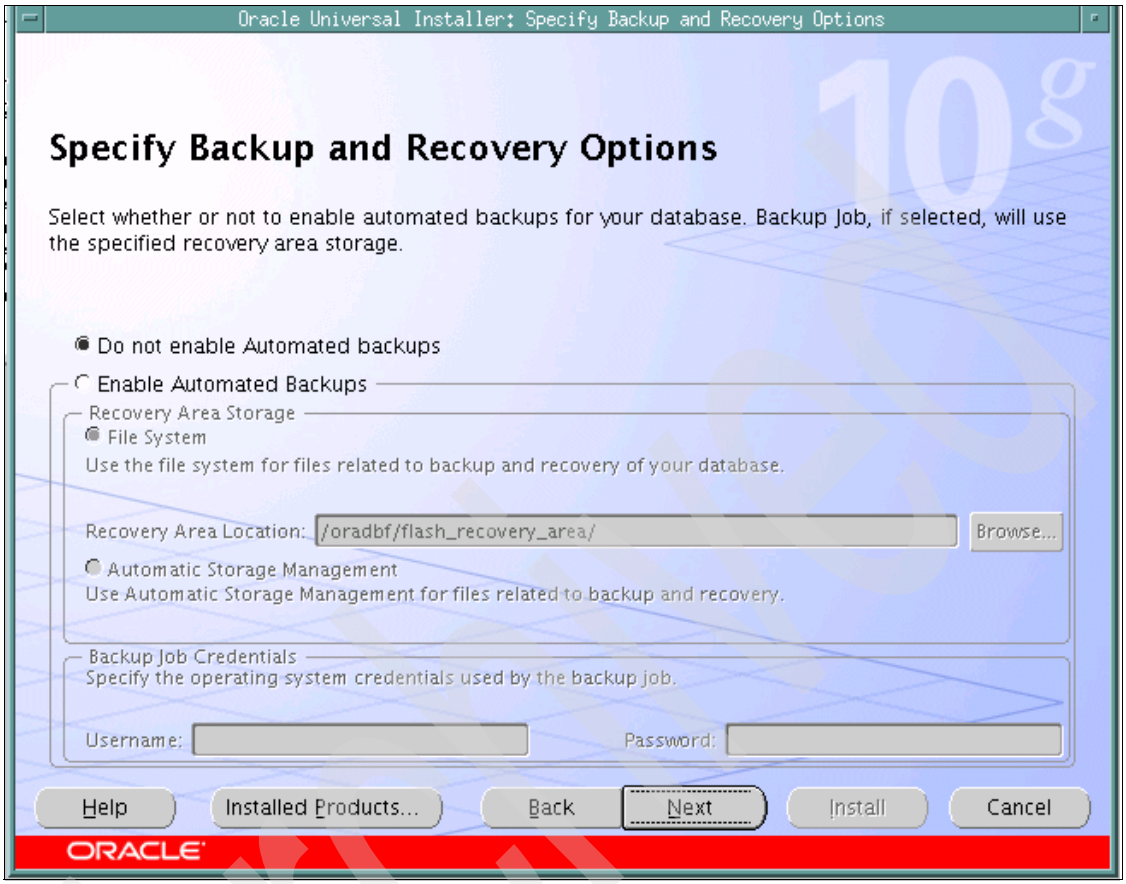


Figure 3-14 Backup and recovery options

14. We chose to use the same password for all Oracle schema. This is not a good idea, for security reasons, on a database that you will use for test, development, or production purposes (Figure 3-15).

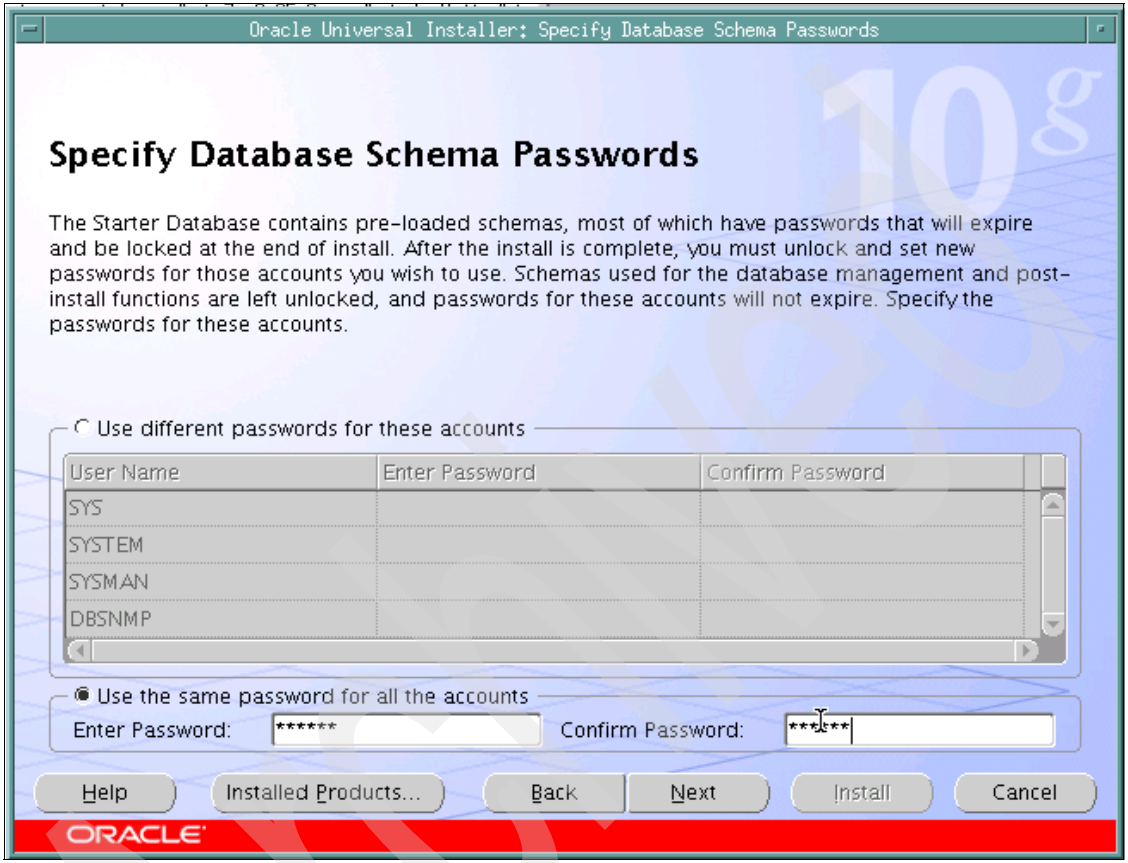


Figure 3-15 Schema passwords

15. A summary of the installation is shown in Figure 3-16.

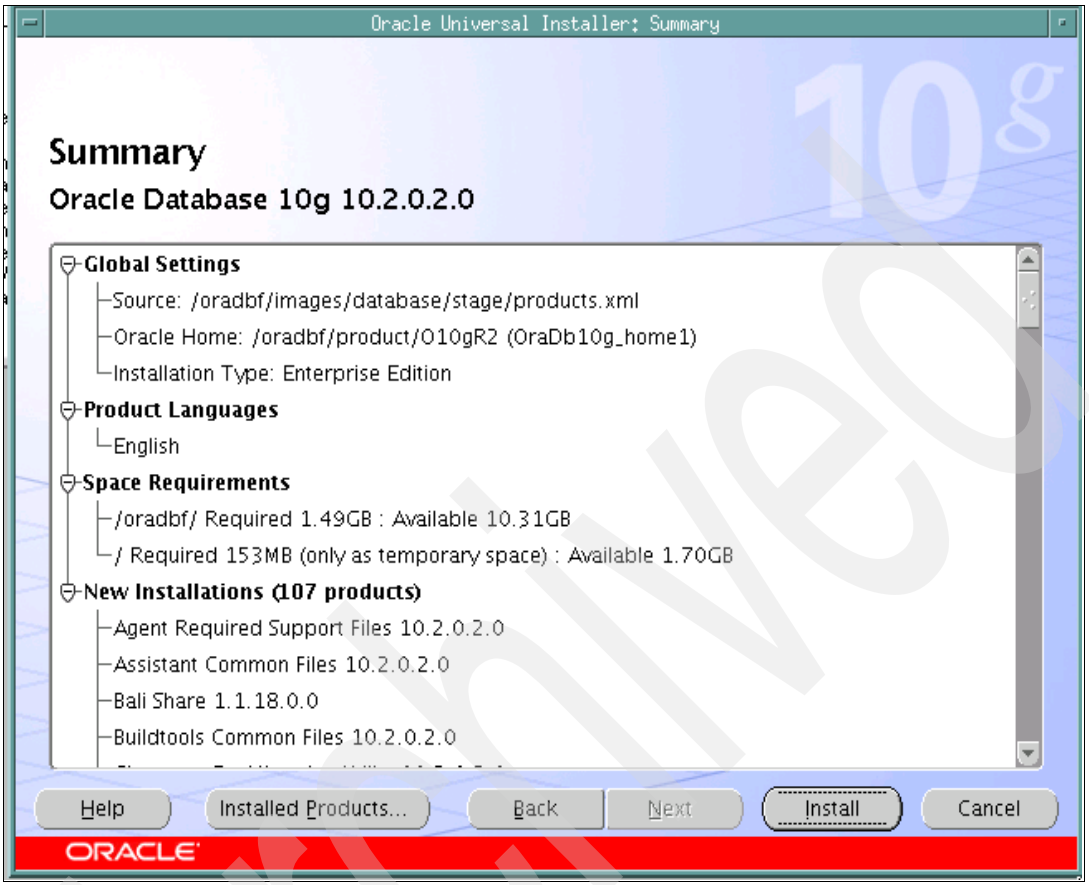


Figure 3-16 Summary of installation

16. The installation process proceeds for the next several minutes. It copies files and runs **make** commands to perform links. See Figure 3-17.

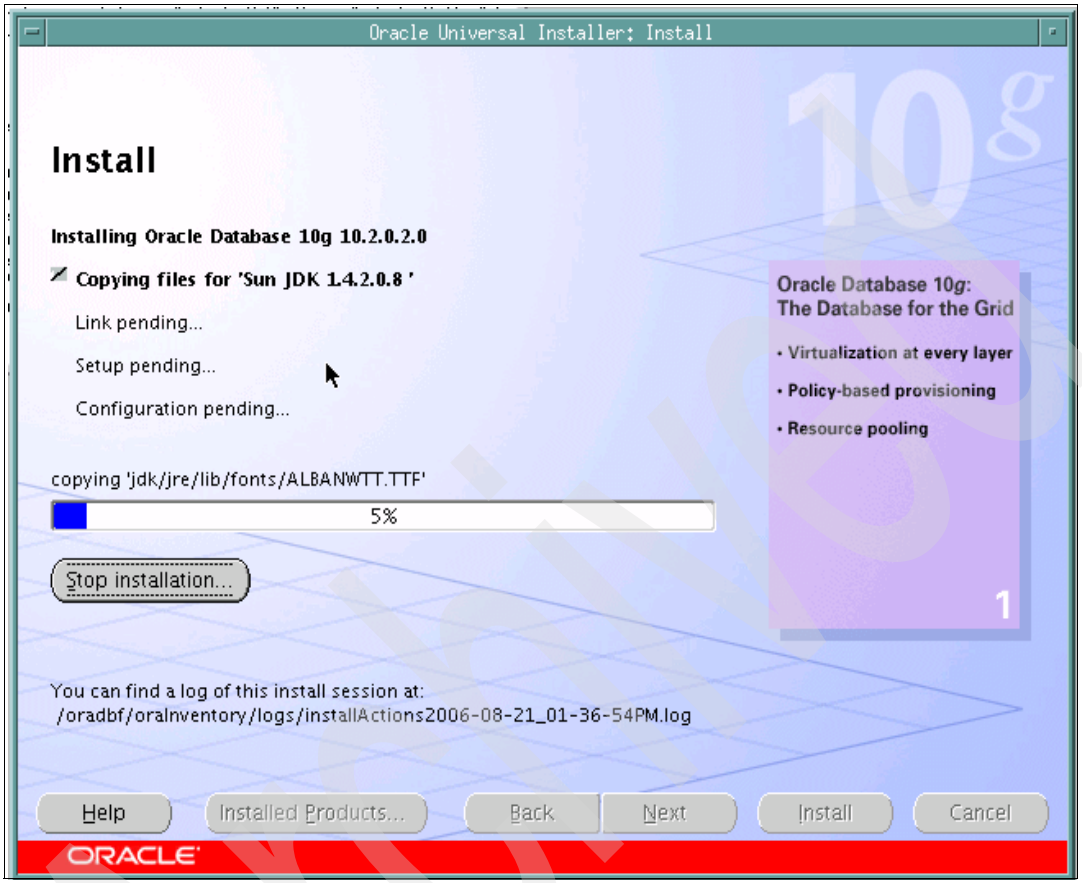


Figure 3-17 Installation of software

17. The installation process automatically starts the agents to configure the network components of the database and creates a database configuration assistant called iSQL*Plus. See Figure 3-18.

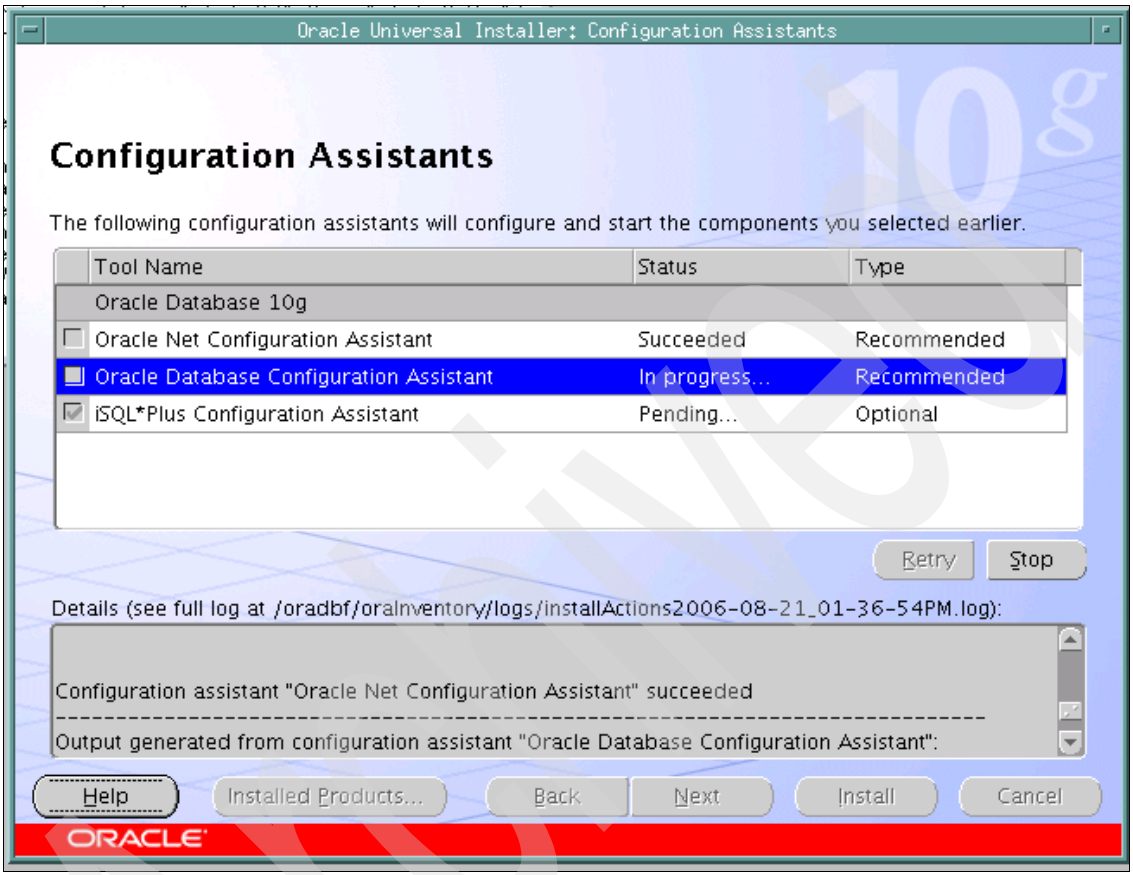


Figure 3-18 Configuration Assistants

18. The database that we configured is created during this process, as shown in Figure 3-19.

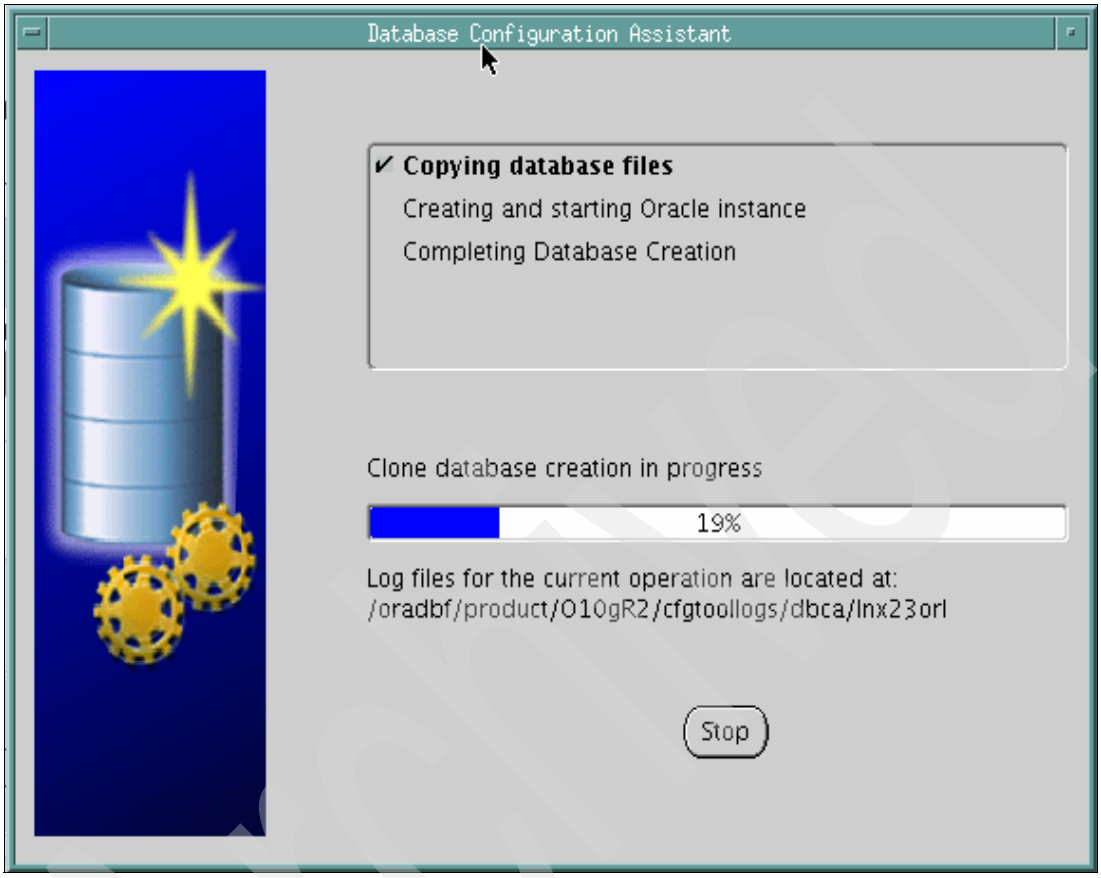


Figure 3-19 Database creation

19. Next, a password and account management window allow passwords for accounts to be changed and various accounts to be unlocked. Selecting **OK** in this window allows the process to continue.

20. Following the instructions on the Execute Configuration scripts window (Figure 3-20), we open a root user PuTTY session.

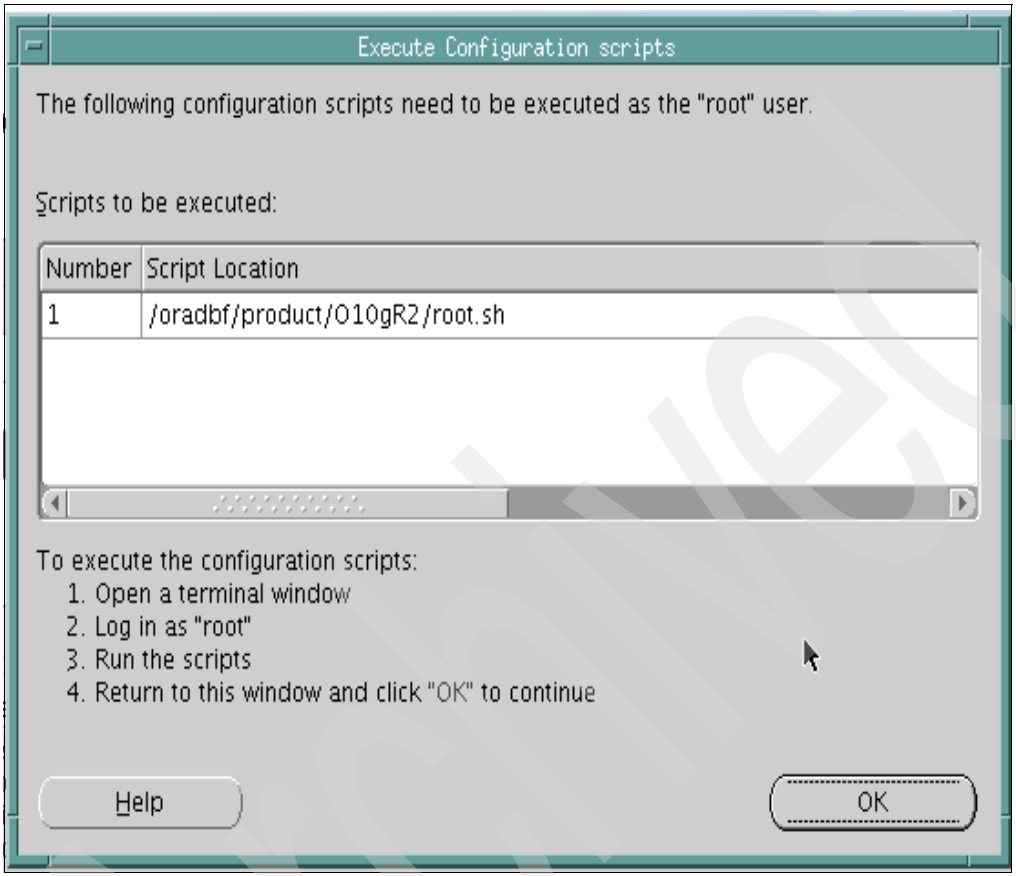


Figure 3-20 Execute Configuration scripts

21. From this root user PuTTY session, we entered the commands shown in Example 3-7 on page 51.

Example 3-7 ls command

```
[root@linux23 bin]# cd /oradbf/product/10gR2
[root@linux23 10gR2]# ls
OPatch       install                          mgw          rdbms
assistants   install.platform                  network      relnotes
bin          inventory                        nls          root.sh
cdata        javavm                           oc4j         slax
cfgtoollogs  jdbc                             olap         sqlj
clone        jdk                              opmn         sqlplus
config       jlib                             oraInst.loc  srvm
crs          jre                              oracore      sysman
css          ldap                             ord          tg4sybs
ctx          lib                              oui          tg4tera
dbs          lib32                            owm          uix
demo         linux23.itso.ibm.com_lnx23orl    perl         wwg
diagnostics  log                              plsql        xdk
has          md                               precomp
hs           mesg                             racg
[root@linux23 10gR2]# ./root.sh
Running Oracle 10g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /oradbf/product/10gR2

Enter the full pathname of the local bin directory: [/usr/local/bin]:
/usr/local
/bin
   Copying dbhome to /usr/local/bin ...
   Copying oraenv to /usr/local/bin ...
   Copying coraenv to /usr/local/bin ...

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
[root@linux23 10gR2]#
```

22.Figure 3-21 shows the successful completion of the installation.

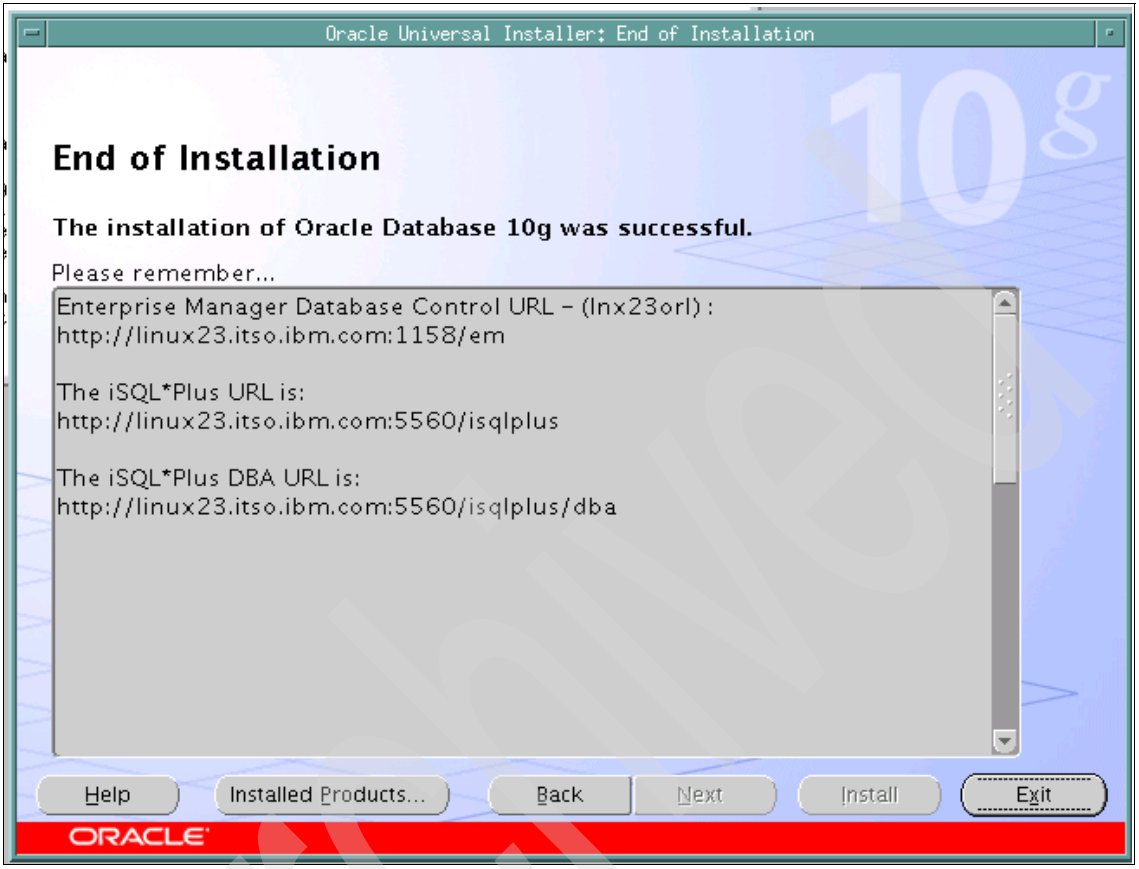


Figure 3-21 End of Installation

23. Clicking **Exit** elicits the exit confirmation prompt, shown in Figure 3-22, to which we reply by clicking **Yes**.

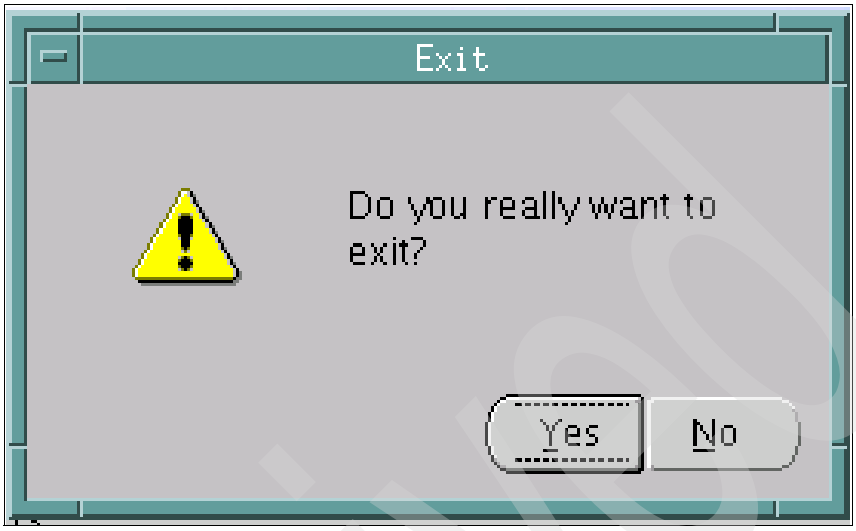


Figure 3-22 Exit confirmation window

24. Figure 3-23 is a confirmation that the database is up.

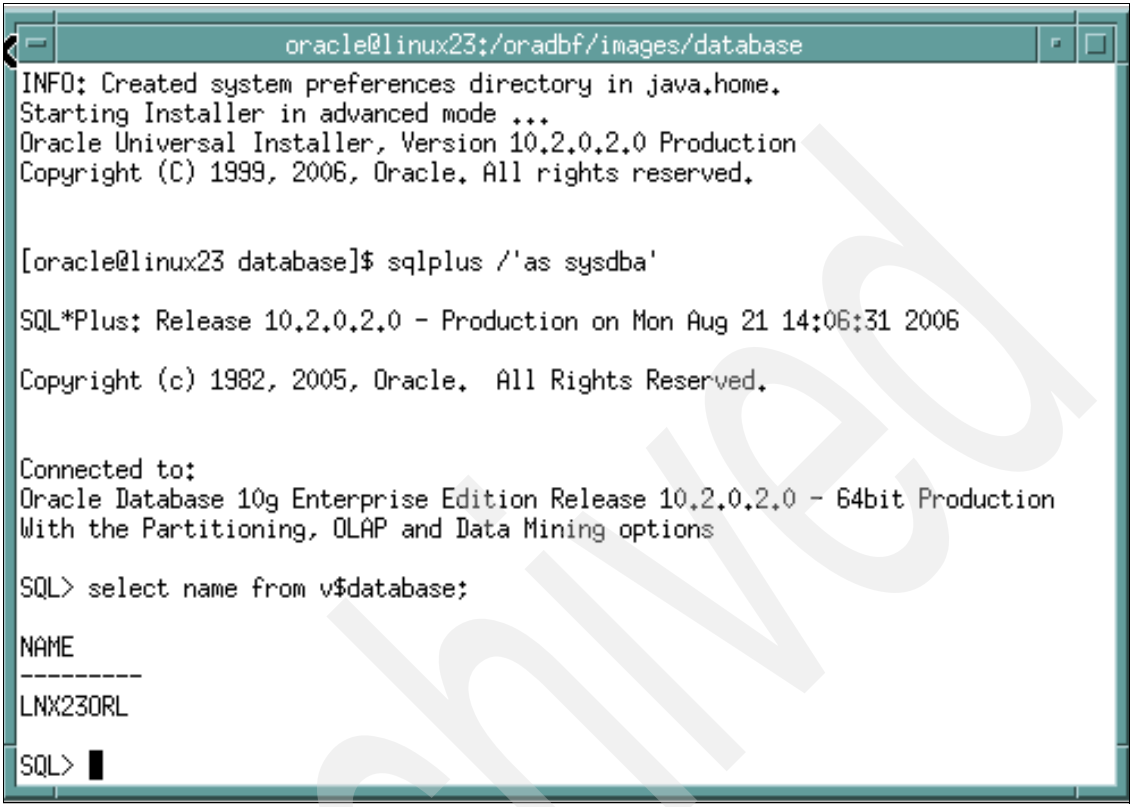


Figure 3-23 Confirmation that the database is up

The Oracle binaries and the Oracle database instance lnx23orl have now been successfully installed. Note the Web sites for accessing the Oracle Enterprise Manager (OEM) and iSQL. See Figure 3-24 on page 56 and Figure 3-27 on page 59 where we used these Web sites.

The Oracle Listener should also be running at the completion of the installation. Enter the command **`lsnrctl status`** and you should get a message indicating that the listener is up for the database. You start and stop the listener by entering the following command:

```
lsnrctl {start, stop}
```

To restart the database, issue the following commands:

```
sqlplus /'as sysdba'
SQL>startup
```

At this point, you have a running database.

Note 1: Our installation of Oracle did not necessarily follow the Oracle Flexible Architecture (OFA) recommendations. The Linux virtual machines we used were used for several projects and we did not have the flexibility to continually change Linux configurations to meet the OFA recommendations. Your installation should follow the best practices used by your organization to install Oracle products. Our installation process does follow the Oracle documentation for process and parameters and as such does indicate how a successful installation would appear.

Note 2: You may want to consider using the *orarun* package, which is a RPM software package provided as part of SUSE Linux Enterprise Server (SLES). The purpose of the orarun package is to simplify the installation and administration of Oracle software products. More information is available at http://wiki.novell.com/index.php/Orarun_package.

3.5 Using OEM to manage an Oracle database

You can use Oracle Enterprise Manager (OEM) through DBControl to manage an Oracle database. After the successful completion of the installation of the database, you can invoke Oracle Enterprise Manager using the Web sites that are displayed by the OUI, as shown in Figure 3-24. The OEM Login window is the result of entering the Web site (*host:port/em*) in an Internet Explorer® browser. You can log on as User Name sys with the password that you entered in the window in Figure 3-24 and connect as SYSDBA in the list box.

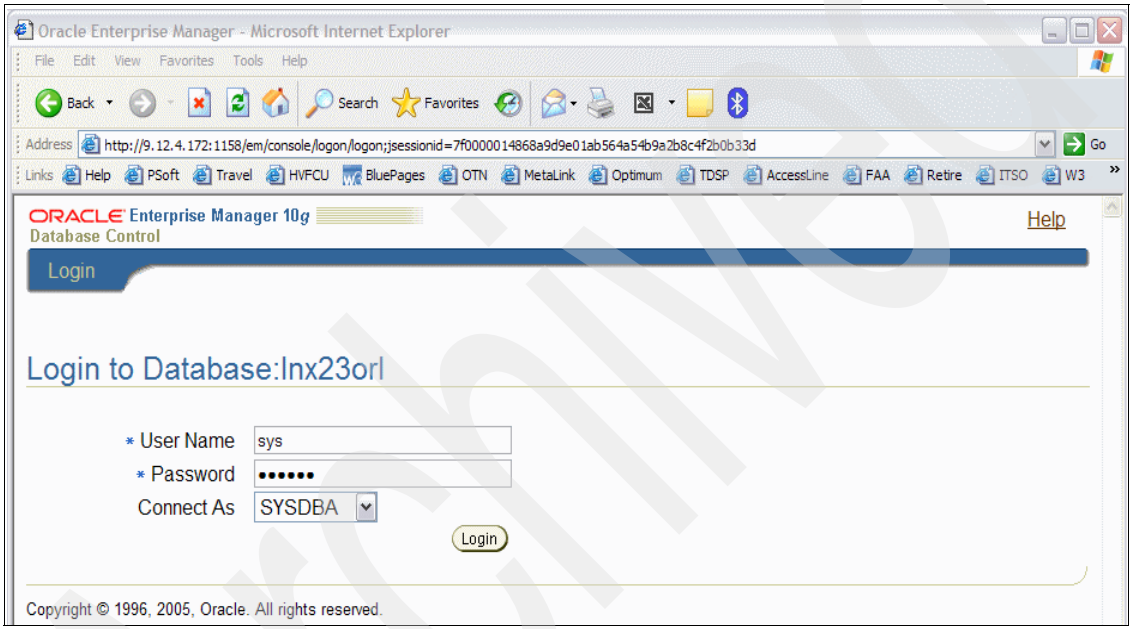


Figure 3-24 Logon window to OEM

The Oracle Enterprise Manager (OEM) Home window displays after the logon is complete (Figure 3-25 on page 57). It is a scrollable panel. This only shows the top part of the functions available.

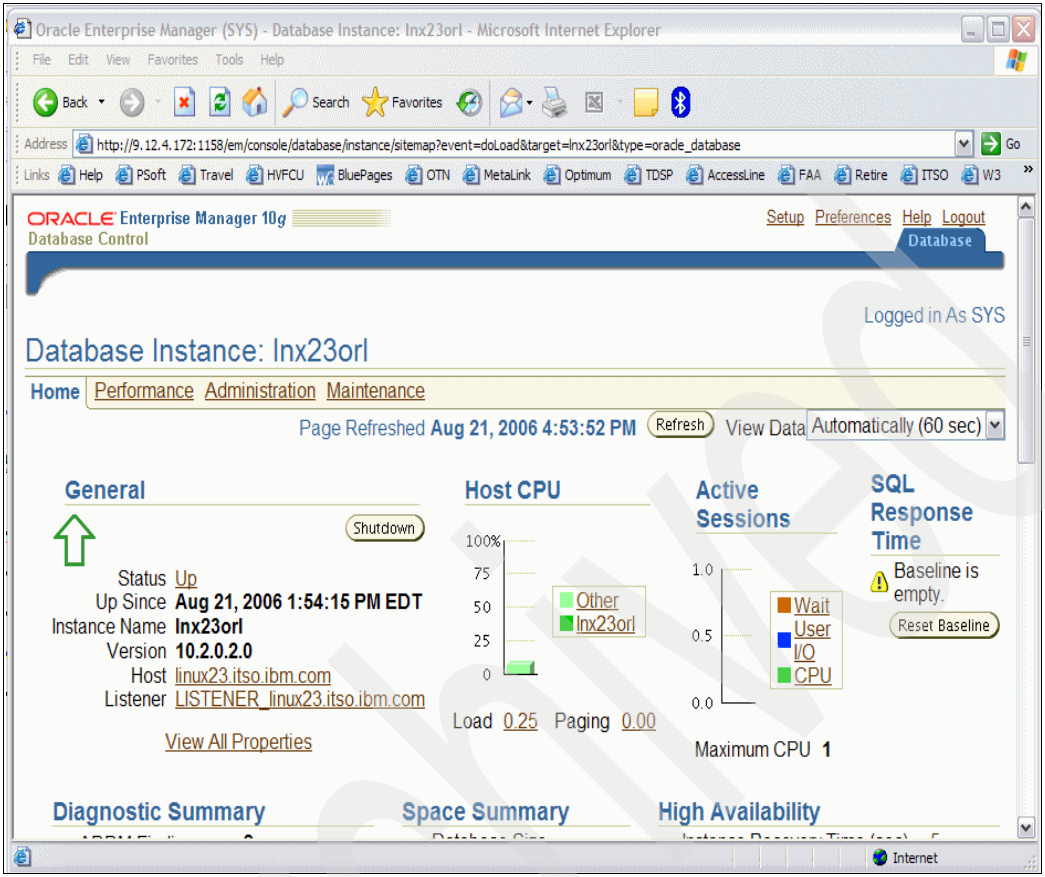


Figure 3-25 Home window for Oracle Enterprise Manager

You can use this window to manage your single instance database, such as checking for patches, running the Automatic Workload Repository, and so on.

3.5.1 Using iSQL*Plus to query a database

Also on the last window of the OUI, there is a Web site to start iSQL*Plus, `http://host_name:5560/isqlplus`. If you enter this in your browser, you get a logon window (Figure 3-26) and then the query window. The system presents you with an SQL worksheet so that you can query the database from your browser. You can also use this to build PL/SQL blocks of code.

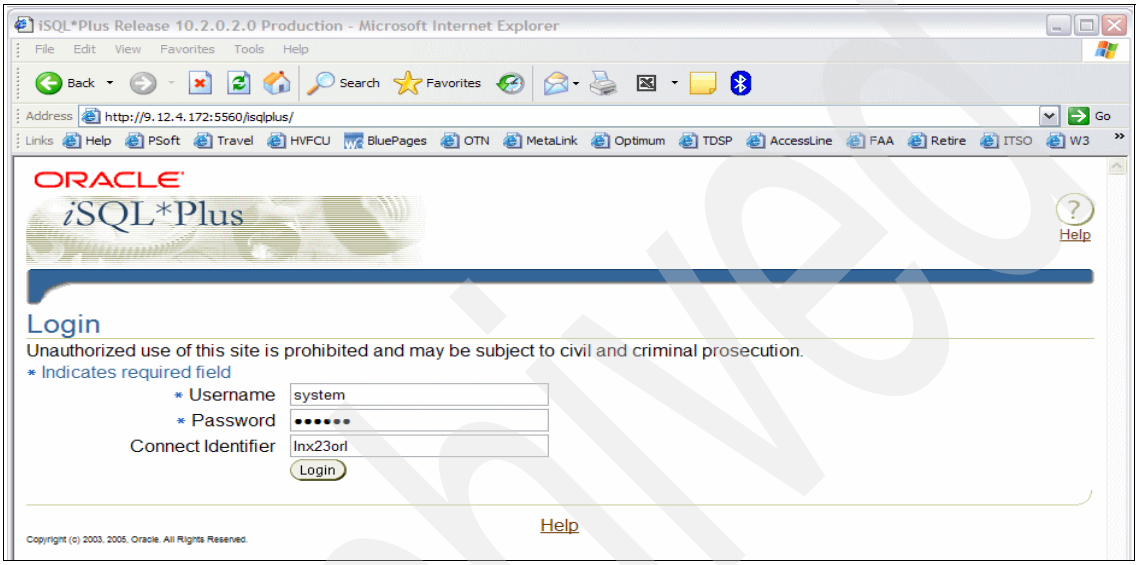


Figure 3-26 iSQL logon window

To ensure that the iSQL*Plus worked, we pointed our browser at the port identified in the last window of the installation. We logged in as system and executed a simple query to ensure that this worked (Figure 3-27).

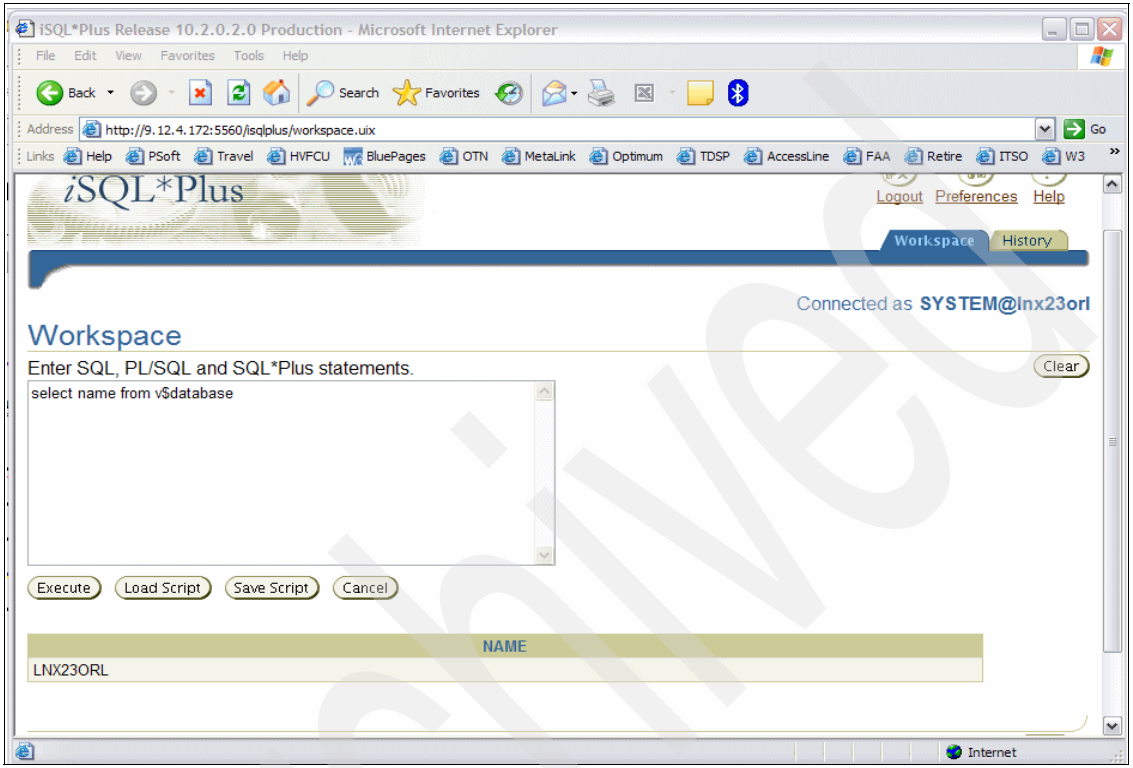


Figure 3-27 Using SQL through a browser

3.6 Summary

The installation using the OUI is very easy and uncomplicated. However, to make sure that the installation is problem-free, it is important that you complete all of the tasks to prepare for the environment correctly. This includes the kernel parameters, and so on. The most important task is to correctly install Linux and ensure that the correct Red Hat Package Managers (RPMs) are installed. If you follow the instructions above to install Red Hat and then the Oracle Installation Guide to ensure that all the necessary RPMs are installed, the installation should be smooth and trouble free.

If you are using SUSE Linux Enterprise Server 9, we used this process for the installation of Linux. We found that if you use this method, you should not have any RPM-related problems installing Oracle.

We used YaST2 to perform a default installation of SUSE Linux Enterprise Server (Figure 3-28).

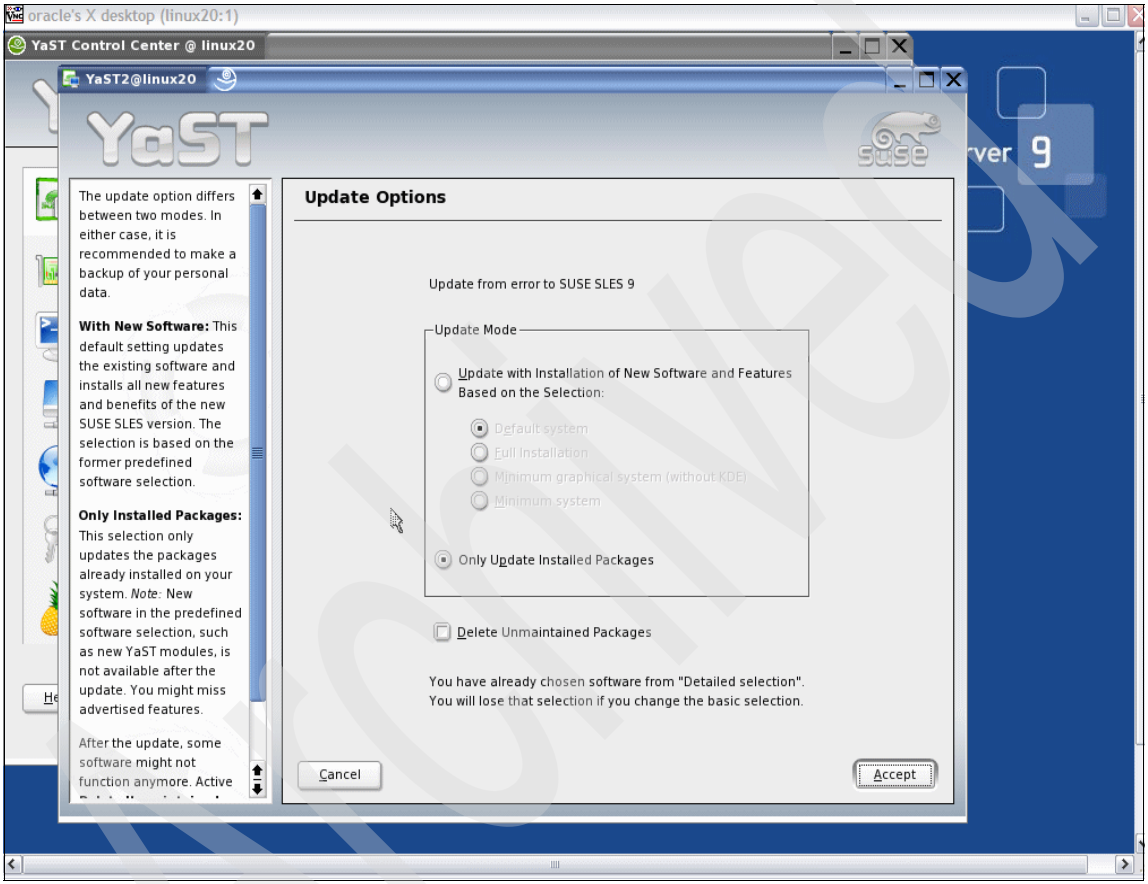


Figure 3-28 Linux default installation

Along with the default installation, you need to install the C/C++ Compiler and Tools package as well. Figure 3-29 demonstrates how we accomplished those tasks.

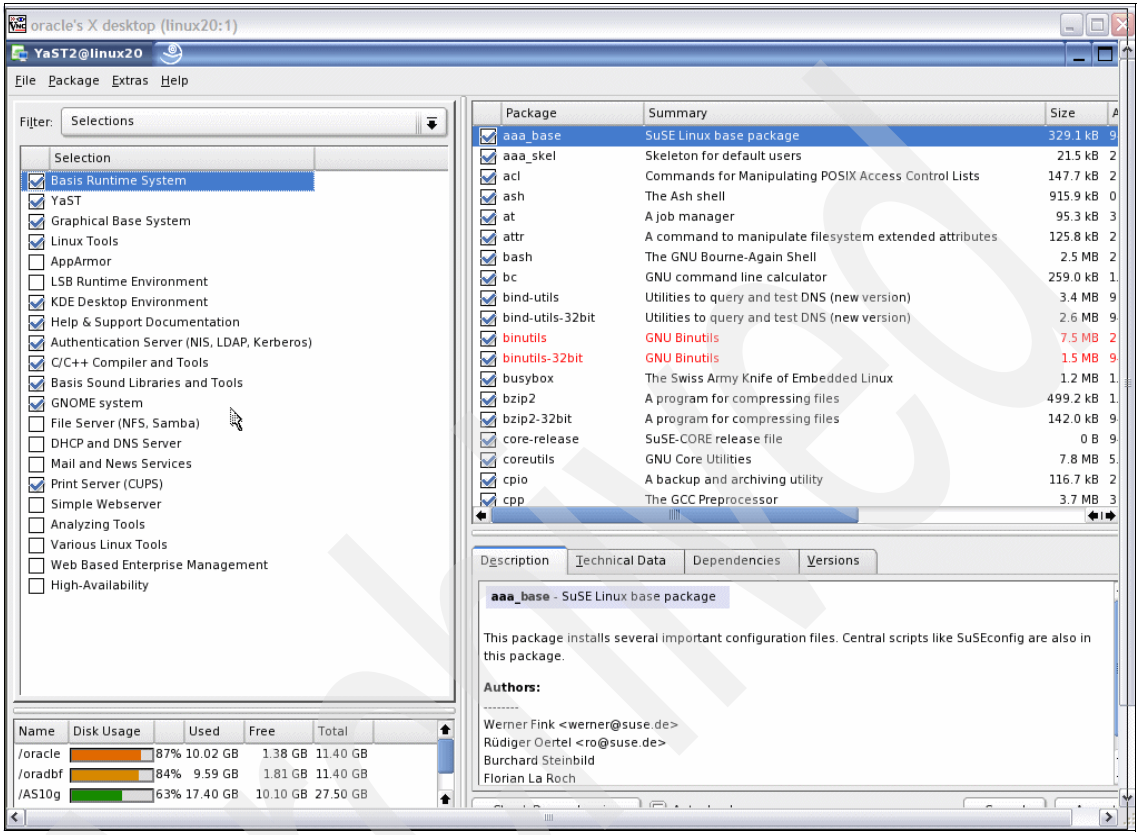


Figure 3-29 Selecting C/C++ compiler and tools

In this chapter, we explained how we installed Linux, either Red Hat or SUSE Linux Enterprise, to ensure that there are no RPM problems encountered when installing Oracle 10gR2. This does not mean that there are not other ways to accomplish this task, but we found that doing it this way ensured our success.

4

Setting up Oracle RAC on Linux for System z

Oracle Real Application Clusters (RAC) is a feature of the Oracle database that provides increased availability and scalability. On System z, the Linux guests using Oracle RAC can be on the same IBM System z9™ or on two z9s.

The Oracle Clusterware, Oracle Binaries for RAC, and a RAC database are easy to install if the infrastructure is correctly set up. This chapter covers the items to consider when setting up the infrastructure. If this setup is successful, the cluster verify utility will work and the installation will be successful. The details of the installation of Oracle 10gR2 RAC are beyond the scope of this book.

In this chapter, we cover:

- IBM System z Network Options for IP interconnect
- IBM System z disk storage options
- Disk formatting options
- Setting up for ASM
- Getting ready for Oracle installation and other sources of information

4.1 Introduction

This document provides an introduction to the various network and disk storage options available for setting up an Oracle RAC system on a System z (a Linux on z/VM system).

This chapter introduces some of the options available for network configuration, disk storage, and performance tuning. Tips for setting up these configurations and best practices will be discussed. References to further detailed guides will also be provided.

4.2 Oracle 10g RAC on IBM System z network options

Oracle 10g RAC under z/VM has various options available for setting up the network infrastructure between nodes. These various network architectures can be on the same Logical Partition (LPAR), separate LPARs, or on separate IBM System z machines within the same sysplex.

An LPAR is a division of processors, memory, and storage into multiple sets of resources, so that each set of resources can be operated independently with its own operating system instance and applications.

The IBM System z platform is the highest availability platform available for customers today. By virtue of moving to a IBM System z platform even without Oracle RAC, a higher level of availability and scalability can be achieved. When selecting the best infrastructure for installing Oracle RAC on IBM System z, the points of failure shown in Table 4-1 should be considered.

Table 4-1 Points of failure

SIngle point of failure	**Probability of failure**	**Cost to fix**
System z hardware	Very Low	High
LPAR	Very Low	Low
z/VM	Low	Low
Linux	Low	Very Low
Disk Subsystem Microcode	Low	Medium
Application	High	Very Low

Every Oracle RAC node requires at least three IP addresses to be assigned. These include a public interface, a private interconnect interface, and an Oracle Virtual IP (VIP) address in order to operate correctly. The Oracle VIP is assigned when Oracle Clusterware is installed and started.

The public/user interface is the standard network interface used for SSH or telnet into the box and is the network interface used to install and configure the Oracle RAC system and to access Oracle Enterprise Manager, if it is installed with the database.

The private interconnect is used for inter-node communication by both Oracle Clusterware and Oracle RAC. This connection needs to be as fast as possible in order to avoid inter-instance cache-fusion related issues. Typically, this private interconnect is a non-routable IP address, such as a 10.x.x.x or 192.x.x.x, since the only communication should be between eligible nodes in the Oracle RAC configuration.

The last required IP address is the Oracle Virtual IP. This is not assigned to any network card by Linux, but rather is configured when the Oracle Clusterware service software is installed and running. The Oracle VIP should also be used when setting up Oracle Client communication to an Oracle RAC node in order to provide proper failover and load balancing capabilities.

Note: The public network node name and the Oracle VIP name must be registered in DNS for the Oracle RAC system to function properly.

Example 4-1 shows a sample **`ifconfig`** output from a running 10g IBM System z RAC system.

Example 4-1 ifconfig output

```
[root@oralin01 ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:00:00:00:00:02
          inet addr:9.82.25.84  Bcast:9.82.25.95  Mask:255.255.255.240
          inet6 addr: fe80::200:0:1c00:2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6142 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5898 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:494424 (482.8 KiB)  TX bytes:1971781 (1.8 MiB)

eth0:1    Link encap:Ethernet  HWaddr 02:00:00:00:00:02
          inet addr:9.82.25.86  Bcast:9.82.25.95  Mask:255.255.255.240
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
hsi0      Link encap:Ethernet  HWaddr 00:00:00:00:00:00
          inet addr:10.10.10.84  Bcast:10.10.10.255  Mask:255.255.255.0
          inet6 addr: fe80::ff:fe00:0/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:1160940 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1122806 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:355623537 (339.1 MiB)  TX bytes:290905100 (277.4 MiB)
```

Example 4-2 shows the /etc/hosts file that was used. (The localhost line needs to be the first line.)

Example 4-2 /etc/hosts file

```
-bash-3.00$ cat /etc/hosts

# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 localhost.localdomain localhost
9.82.25.84 oralin01 oralin01.wsclab.washington.ibm.com
9.82.25.85 oralin02 oralin02.wsclab.washington.ibm.com
9.82.25.86 oralin01-vip oralin01-vip.wsclab.washington.ibm.com
9.82.25.87 oralin01-vip oralin02-vip.wsclab.washington.ibm.com
10.10.10.84 oralin01hs oralin01hs.wsclab.washington.ibm.com
10.10.10.85 oralin02hs oralin02hs.wsclab.washington.ibm.com
```

Example 4-3 shows an example of a configured tnsnames.ora file that allows for load balancing to an Oracle RAC system utilizing the Oracle VIP address.

Example 4-3 tnsnames.ora entry

```
# Generated by Oracle configuration tools.

RACTST =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = oralin01-vip)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = oralin02-vip)(PORT = 1521))
      (LOAD_BALANCE = YES)
    )
    (CONNECT_DATA =
      (SERVICE_NAME = ractstXDB)
      (FAILOVER_MODE =
        (TYPE = session)
        (METHOD = basic)
```

```
        (RETRIES = 20)
        (DELAY = 5)
      )
    )
  )
```

Depending on your availability requirements and system configuration, you can set up your network interfaces in several different ways, such as using a Virtual Switch (VSwitch), a HiperSocket, or by utilizing a Gigabit Open System Adapter (OSA). Figure 4-1 illustrates some of the network topologies that Oracle RAC can be configured with.

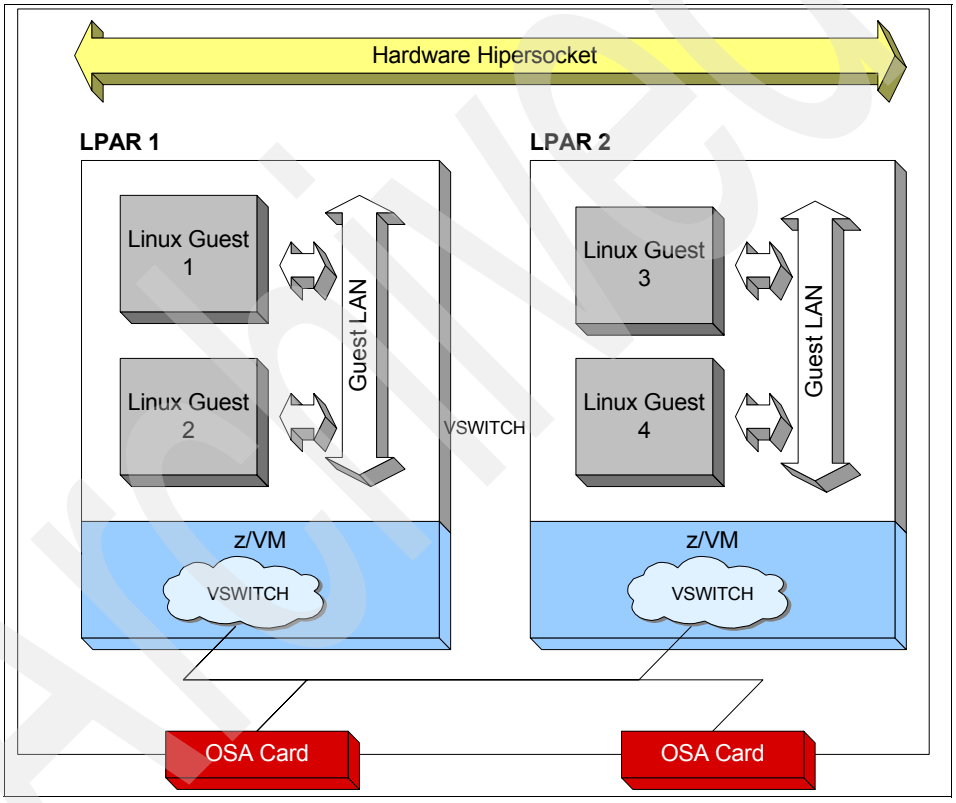


Figure 4-1 Network topology options

4.2.1 Using a Virtual Switch

A Virtual Switch (VSwitch) bridges real hardware and virtual networking LANs, offering external LAN connectivity to the guest LAN environment. A VSwitch can be used within the same LPAR, connecting multiple z/VM guests using virtual network adapters rather than dedicated hardware network adapters.

If two or more Oracle RAC nodes are on the same LPAR, then a guest local area network can be used for communication between the RAC nodes. A guest LAN is a virtual LAN connection between virtual machines that use either the z/VM simulation of HiperSockets™ (without the need for HiperSockets microcode) or a virtual OSA-Express (QDIO) adapter.

When setting up the Oracle private interconnect between Linux guests on the same LPAR, a HiperSocket Guest LAN can be used. For the main Oracle public interface, where broadcast and routing capabilities are needed, the OSA or QDIO emulation should be used.

The deployment of a virtual switch reduces the CPU utilization cost and latency associated with providing external connectivity through a physical router. The switching logic resides in the z/VM Control Program (CP), which owns the connection and performs all data transfers between guest LAN nodes. This in turn helps eliminate the impact associated with a router.

4.2.2 Using HiperSockets

HiperSockets enables you to quickly transfer data between operating system instances running on different LPARs. On system z, HiperSockets supports up to 16 independent virtual LANs that operate as TCP/IP networks within a mainframe's central electronics complex (CEC).

HiperSockets achieves high-speed TCP/IP connectivity between servers on separate LPARs by using a subchannel on the shared HiperSocket (IQD chpid). This subchannel then communicates with another host using the same chpid (that is, the communication is through the system memory of the processor).

The speed of HiperSockets is directly related to the speed of processors on the CEC, because a HiperSocket network is implemented by microcode that runs on the CEC processors.

HiperSockets is an in-memory network connection. By keeping the network packets local to the mainframe, the network configuration becomes much simpler and therefore reduces the networking logistics and external cabling. Compared to OSA, HiperSockets is quite fast.

4.2.3 Gigabit Open System Adapter (OSA) card

On the IBM System z machine, an OSA-Express/OSA-Express2 Gigabit Ethernet can be used for the Oracle RAC interconnect as the network card used between two or more RAC nodes. This can also be achieved through a VSwitch emulating an OSA card.

An OSA card is typically used when two or more RAC nodes are on separate IBM System z machines, but can also be used as an alternative if HiperSockets, for example, is not available between LPARs.

OSA-Express and OSA-Express2 cards support a mode of operation called Queued Direct I/O, or QDIO. QDIO is a highly efficient data transfer architecture that dramatically improves data transfer speed and efficiency for TCP/IP traffic.

4.2.4 Best practices for network configuration

If you are setting up a test or development system on one Linux LPAR, utilizing a VSwitch for both the public and private interfaces is recommended. The Oracle private interconnect can use a HiperSockets Guest LAN (this emulates a HiperSocket) and the public network interface can use a Gigabit Ethernet Guest LAN configuration.

For an Oracle RAC system on two or more separate LPARs on the same IBM System z machine, HiperSockets is recommended for the private network interconnect.

When setting up Oracle RAC between two or more separate System z machines, a gigabit OSA Card (or a VSwitch that emulates a gigabit OSA card) is the only configurable option for the private interconnect.

Table 4-2 shows some of these recommendations.

Table 4-2 Network configuration options

Setup	Oracle private network (interconnect)	Oracle public network
All z/VM Linux guests in one LPAR	VSwitch recommended Gigabit OSA Card possible	VSwitch recommended Shared or dedicated OSA Card
z/VM Linux guests on different LPARs	HiperSocket recommended Gigabit OSA card possible	VSwitch recommended Shared or dedicated OSA Card

Setup	Oracle private network (interconnect)	Oracle public network
z/VM Linux guests on different physical machines	GIgabit OSA card	OSA card

4.3 Oracle 10g RAC on IBM System z disk storage options

The IBM System z platform supports many disk configurations for installing and running a 10g Oracle RAC system under z/VM Linux. This section will explore some of these storage options as well as providing some analysis on which options may fit your environment best.

4.3.1 z/VM disk options

One important consideration when setting up an Oracle database on IBM System z is disk storage. It is important for the DBA to understand the underlying disk infrastructure in order to help tune and maintain the system efficiently.

With z/VM Linux, there are two types of disk storage that are typically used. Disk applications can use either SCSI storage devices utilizing fixed (512-byte) blocks or Extended Count Key Data (ECKD) disk devices. These disks can then be connected by the following connection technologies: Enterprise System Connection (ESCON®), Fiber Connector (FICON®), or Fibre Channel Protocol (FCP).

FCP/SCSI devices can be faster than FICON/ECKD devices. One approach for optimal storage is to put large database files on FCP/SCSI disks, and the Linux system and Oracle binaries on the FICON/ECKD devices.

You cannot install Linux on an all-FCP/SCSI partition. System z requires at least one ECKD disk to install the operating system.

With FICON/ECKD storage, you do not need to set up multipathing on the Linux guest, as it can be provided by the channel subsystem with multiple channel path groups. When using FCP/SCSI disk, we recommend implementing DM multipathing to provide the necessary fault tolerance. In order for Oracle ASM to utilize multipathing, the DM multipath devices must be specified (as opposed to the FCP/SCSI device) when assigning the disk in Oracle.

Another consideration with FCP/SCSI disks is FCP channels. With Oracle RAC, the SAN LUNs can be shared across Linux guests, but each guest must use a different FCP channel (or group of FCP channels if DM multipathing is implemented). This may cause concerns if there is any limitation on the number of FCP channels that are available.

Customers implementing FCP disks may also want to consider N-Port Virtualization. With N-Port Virtualization, LUNs can be assigned directly to Linux guests rather than all the guests running under VM. N-Port Virtualization is first supported on the IBM System z9, and not on the z990 or z980 models. See *Fibre Channel Protocol for Linux and z/VM on IBM System z*, SG24-7266 for further details.

When using SAN/SCSI storage, it is very important to use disks from several RAID arrays and from all clusters. This will provide better I/O concurrency and load balancing. Figure 4-2 illustrates the correct way to assign disks. The incorrect way is to allocate all the storage in order from one rank in sequential order, for example, 5100 to 510F.

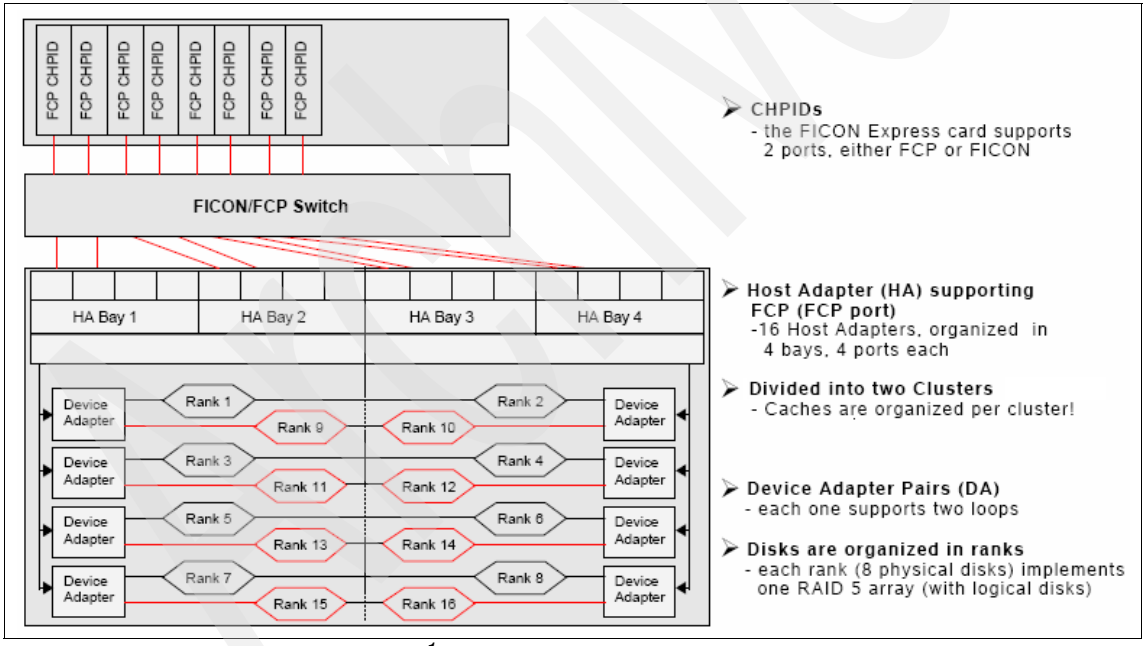


Figure 4-2 Correct way to allocate disk[1]

[1] Source: *Performance experience with Databases on Linux for IBM System z*, by Kammerer, found at http://www.ibm.com/developerworks/linux/linux390/perf/index.html.

Linux's Logical Volume Management (LVM) and Oracle's Automatic Storage Management (ASM) are both disk aggregate stripping and mirroring utilities. You should use one or the other when setting up storage for an Oracle database.

When using LVM, the following recommendation for striping should be used:

- FICON and FCP: The number of stripes should equal the number of disks.
- ESCON: The number of stripes should at least equal the number of available channels.
- Good stripe sizes are 32 KB and 64 KB for data warehouse databases, and 16 KB or 8 KB for high online transaction processing databases (OLTP).

An Oracle ASM database uses 1 MB stripe sizes for data files, and a 128 KB stripe sizes for redo files. If using hardware RAID, make sure the LUN stripe size is as close to 1 MB as possible for the data file LUNs and 128 KB for the redo log LUNs.

You can use the **lsdasd** command to see what devices are assigned to the Linux guest system. See Example 4-4.

Example 4-4 Listing DASD assigned to a Linux guest

```
[root@oralin01 ~]# lsdasd
0.0.0200(ECKD) at ( 94:  0) is dasda      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0201(ECKD) at ( 94:  4) is dasdb      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0202(ECKD) at ( 94:  8) is dasdc      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0203(ECKD) at ( 94: 12) is dasdd      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0400(ECKD) at ( 94: 20) is dasdf      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0401(ECKD) at ( 94: 24) is dasdg      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0402(ECKD) at ( 94: 28) is dasdh      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0403(ECKD) at ( 94: 32) is dasdi      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0404(ECKD) at ( 94: 36) is dasdj      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0405(ECKD) at ( 94: 40) is dasdk      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0406(ECKD) at ( 94: 44) is dasdl      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0407(ECKD) at ( 94: 48) is dasdm      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0408(ECKD) at ( 94: 52) is dasdn      : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0500(ECKD) at ( 94: 56) is dasdo      : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0501(ECKD) at ( 94: 60) is dasdp      : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0502(ECKD) at ( 94: 64) is dasdq      : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0503(ECKD) at ( 94: 68) is dasdr      : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0504(ECKD) at ( 94: 72) is dasds      : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0505(ECKD) at ( 94: 76) is dasdt      : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0506(ECKD) at ( 94: 80) is dasdu      : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0507(ECKD) at ( 94: 84) is dasdv      : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0508(ECKD) at ( 94: 88) is dasdw      : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0509(ECKD) at ( 94: 92) is dasdx      : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.050a(ECKD) at ( 94: 96) is dasdy      : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.050b(ECKD) at ( 94:100) is dasdz      : active at blocksize 4096, 601020 blocks, 2347 MB
```

```
0.0.050c(ECKD) at ( 94:104) is dasdaa     : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.050d(ECKD) at ( 94:108) is dasdab     : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.050e(ECKD) at ( 94:112) is dasdac     : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.050f(ECKD) at ( 94:116) is dasdad     : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0510(ECKD) at ( 94:120) is dasdae     : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0511(ECKD) at ( 94:124) is dasdaf     : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0512(ECKD) at ( 94:128) is dasdag     : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0513(ECKD) at ( 94:132) is dasdah     : active at blocksize 4096, 601020 blocks, 2347 MB
0.0.0300(FBA ) at ( 94:136) is dasdai     : active at blocksize 512, 614400 blocks, 300 MB
0.0.0301(FBA ) at ( 94:140) is dasdaj     : active at blocksize 512, 614400 blocks, 300 MB
0.0.0302(ECKD) at ( 94:144) is dasdak     : active at blocksize 4096, 360000 blocks, 1406 MB
```

4.3.2 Setting up disks between separate z9 machines

Typically, when setting up disks between z9 machines, the first Linux guest node owns the disks in read/write mode, while the second guest links to the disks in read/write.

Because the Linux guests are on two different z9 machines, the directory entries for both virtual machines will use dedicate statements to define the devices being shared. For example, the first node would have DED 0340 1F91 in the directory entry to define the virtual 340 device, and the second node would have DED 0340 2F91. It is important to let CP know that the physical devices are shared with another system. This is done through the system config file and an Rdevice statement. The following is an example:

```
Rdevice xxxx-yyyy Type DASD Shared Yes MDC OFF
```

In this statement, xxxx-yyyy is the range of real device addresses that are shared between the two z9 machines. Because the devices are shared, we recommend making sure that each system does not cache any of the I/Os, so you should use MDC OFF. Having the devices dedicated means that they are ineligible for a minidisk cache.

Dedicating a device to a virtual machine means that it is not available for use by any other virtual machine under that z/VM image. It is not possible to mix minidisks and dedicated devices. In order to use minidisks, a device needs to be attached to the system. When a device is dedicated, it is attached to a particular virtual machine.

4.3.3 File system setup

Oracle under z/VM Linux has various options available for setting up the storage needed to run Oracle RAC. Several file system groups are required to run Oracle. These include:

- Standard operating system disk(s) for each node with SUSE/Red Hat
- Swap disk for each node in the cluster
- Local disk for the Oracle Binaries, patches, and Oracle logs on each node
- Shared disk between the Oracle RAC nodes for database files

The standard operating system disk will require an additional 400 MB free under /tmp for Oracle to install. Typically, this should be a size of at least 2.1 GB.

Figure 4-3 shows a sample RAC file system devices setup.

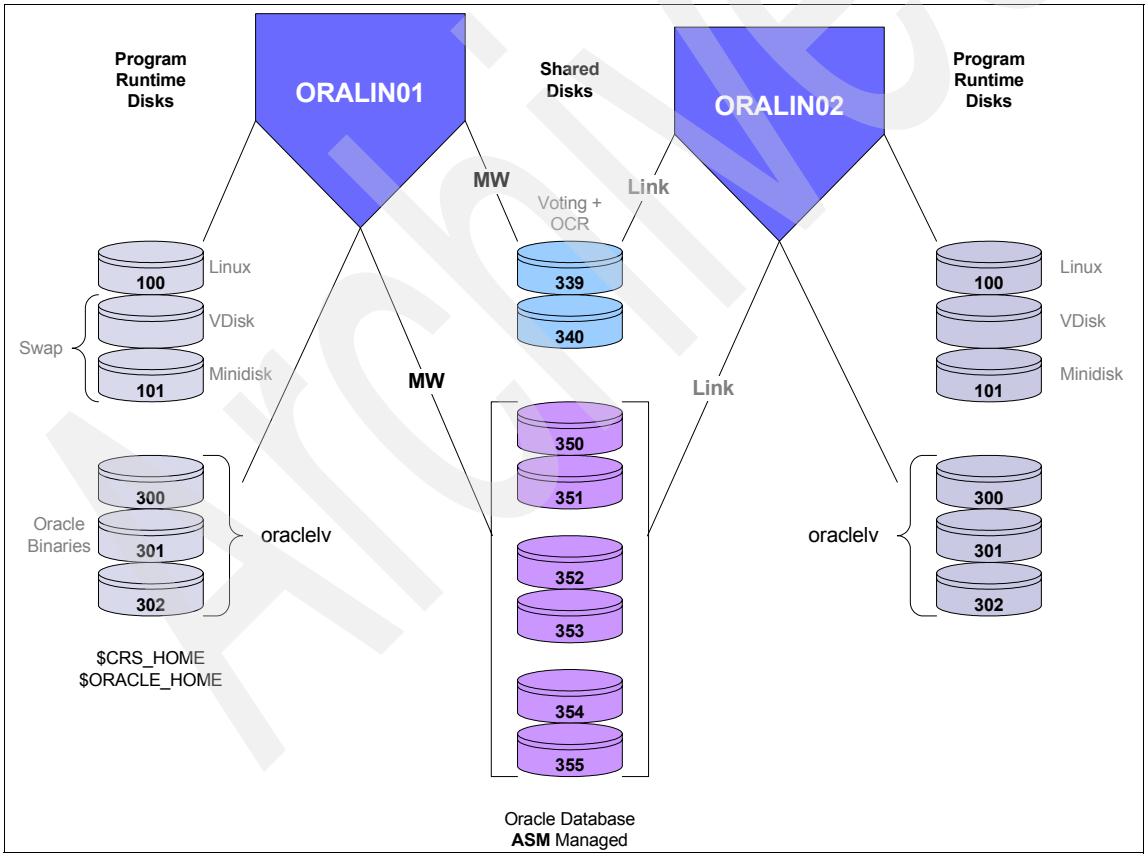


Figure 4-3 Sample RAC file system device setup

The first Linux guest node owns the disks in read/write mode, while the second guest links to the disks in read/write mode.

4.3.4 Linux swapping recommendations

Sizing Linux swapping on z/VM environments correctly is very important, or the performance of the system will diminish.

Setting up a small virtual disk in storage (VDISK) and a minidisk with the higher priority on the VDISK can provide a method for sizing memory for Oracle and the Linux virtual machine. If there is paging to the second paging disk, then you should consider adding memory to Linux or reduce the Oracle SGA/PGA. Some swap to the VDISK is okay, but extensive swapping to the minidisk is a sign that memory is not sized correctly.

See 5.3.2, “Swap space for Linux” on page 92 for more swap space recommendations.

4.3.5 Oracle binaries installation

When installing Oracle RAC on IBM System z, several independent sets of Oracle binaries must be installed for Oracle Clusterware and the Oracle database software to function properly. From an Oracle perspective, these are typically called ORACLE_HOME and CRS_HOME.

You must install the Oracle Clusterware in a separate Oracle Home, which typically uses about 1.5 GB of storage. The Oracle Home for the RAC database will need another 2.0 GB of storage. Oracle also advises that when using Oracle ASM disk storage (which is fairly typical for Oracle RAC) to have another independent Oracle Home for it.

The requirement of another Oracle home for ASM binaries is not mandatory, but makes things easier for database patching since the Oracle ASM set of binaries can be patched while the database is running, reducing the required downtime for the node.

Space is also needed for the alert logs, trace files, database patches, and the Oracle install media zip files, so typically a size of 15-20 GB is used on each node for installing the Oracle software.

4.3.6 OCR and Voting disks installation

Oracle supports two ways of storing the Oracle Clusterware files: either with Raw/Block devices or a shared file system such as Oracle Cluster File System 2 (OCFS2). OCFS2 is an extent based, clustered file system that can be accessed and written to by all nodes in the cluster. OCFS2 is supported on Linux 2.6 kernels and above only.

Oracle Clusterware requires at least two shared devices: one for the Oracle Cluster Registry (OCR) and another for the Voting disks. If you are formatting ECKD type DASDs to use as Raw or Block partitions, you must format these with a 4 KB block size.

For test systems, one partition for the OCR and Voting disks is typical. For production systems, redundant OCRs and multiple Voting disks (must be an odd number) are recommended.

- Two OCRs: 256 MB each or 512 MB
- Three Voting: 256 MB each or 768 MB

Oracle ensures that devices contain a minimum of 256 MB of available disk space. Hence, using at least 280 MB for each device size is recommended to account for the impact of partition creation.

4.3.7 Shared database disk installation

Oracle RAC can be created using the following storage options:

- Oracle ASM for the data files with the OCR and Voting disks on Raw or Block devices
- Standard Raw or Block devices (no Oracle ASM)

It is a good idea to set up multiple disk groups for the different types of I/O that Oracle uses. For example, one set of disks for the database files and another set of disks for redo logs.

If creating an Oracle ASM database, Oracle recommends putting the Flash Recovery Area on a different disk group than the database files. This protects the system from having the active data and backup and recovery data on the same physical disk.

4.4 Formatting and partitioning disks for Oracle RAC

The formatting tool **dasdfmt** provides options to format DASD devices with different hard block sizes. A **dasdfmt** block size of 4096 should be used on z/VM systems:

```
# dasdfmt -f /dev/dasda -b 4096 -p
# dasdfmt -f /dev/dasdb -b 4096 -p
# dasdfmt -f /dev/dasdc -b 4096 -p
```

The next step is to create a partition on each of the disk devices. You should create only one partition on each disk, but for test environments some people create more than one partition on a disk. The commands to create the partitions are similar whether you are using ECKD disk (**fdasd**) or SCSI/FCP disk (**fdisk**):

```
# fdasd -a /dev/dasda
# fdasd -a /dev/dasdb
# fdisk  /dev/sda
```

For Oracle RAC, you will first need to format and partition the OCR, Voting, and any Oracle ASM database disks.

For the non database file systems, you can use the **pvcreate** command to create a physical volume. The **pvcreate** command writes a volume group descriptor at the start of each disk or partition.

Hence, if you wanted to create a file system involving three disks, you would first enter a command similar to the following to mark each device that you want to use in the volume group as a physical volume:

```
# pvcreate /dev/dasda1 /dev/dasdb1 /dev/dasdc1
```

Then the next step is to create a volume group for the physical volumes:

```
# vgcreate orabinvg /dev/dasda1 /dev/dasdb1 /dev/dasdc1
```

To display the attributes of this volume group, you can use the following command:

```
# vgdisplay orabinvg
```

The next step is to create logical file system that can be mounted and used by Oracle.

In this example, we are using three disks and a LV of three stripes, with a block size of 32 KB and a 6.8 GB LV size:

```
# lvcreate -i3 -I32 -L6.8G -norabinlv  orabinvg
```

To display the attributes of the newly created LV, run:

```
# lvdisplay /dev/orabinvg/orabinlv
```

To create an ext3 file system for the oracle logical volume, you need to run **mkfs**. An ext3 file system type is recommended:

```
# mkfs.ext3 /dev/orabinvg/orabinlv -b 4096
```

Create the mount points for Oracle home and CRS home, mount the file system, and verify the size of the newly created file system:

```
# mkdir -p /oracle
# mount /dev/orabinvg/orabinlv /oracle
# df -h
```

To make the changes permanent, you need to update the fstab file. We recommend making a backup of this file first:

```
#vi /etc/fstab
/dev/orabinvg/orabinlv  /oracle                 ext3    defaults
1 2
```

4.5 Installing Oracle ASM Database On raw/block devices

Oracle Automated Storage Management (ASM) is new feature in Oracle 10g that provides the DBA with the ability to manage many database files through the creation of disk groups. A disk group is a set of disks that Oracle ASM owns and manages. By managing just a few disk groups, as opposed to potentially several thousand data files, database management can be made easier.

Oracle ASM also can increase throughput and performance for the database through its ability to stripe table spaces across several disk devices.

Only one ASM instance is required per node regardless of the number of database instances on the node. For z/VM production Oracle environments, we recommend only one ASM and one RAC instance per node so that memory management can be tuned properly for the instance.

For multipath devices, such as /dev/mapper/, you should use the multipath device and not the SCSI device name, such as /dev/sda or /dev/sdb.

As of the Linux 2.6 kernel, it is now recommended to use block devices a opposed to raw devices for the OCR, Voting, and ASM disks.

If using raw devices, you need to update the /etc/raw (SUSE) or /etc/sysconfig/rawdevices (Red Hat) files with the device bindings. The following is an example of the rawdevices file:

```
# This file and interface are deprecated.
# Applications needing raw device access should open regular
# block devices with O_DIRECT.
# raw device bindings
# format /dev/ <rawdev> <major> <minor>
# <rawdev> <blockdev>
# example /dev/ /dev/raw/raw1 /dev/sda1
# /dev/raw/raw2 8 5
/dev/raw/raw1 /dev/dasdo1
/dev/raw/raw2 /dev/dasdp1
/dev/raw/raw3 /dev/dasdq1
```

You must then restart the raw service as the root user, or reboot the server in order to pickup the bindings:

```
# service rawdevices restart
#/etc/init.d # chkconfig -s raw 35 (for Suse)
#/etc/init.d # chkconfig -add rawdevices -- level 35 (for Red Hat)
```

If using block devices, you do not need to update the /etc/raw or /etc/sysconfig/rawdevices files.

In order to assign the correct file permissions for these devices, you need to utilize **udev** to set the file permissions. For Red Hat Enterprise Linux 4 systems, you should create a 49-oracle.permissions file, and for Red Hat Enterprise Linux 5 and SUSE Enterprise Server 10 systems, you should create a 51-oracle.permissions file in the /etc/udev/permissions.d directory.

If using block devices, the file should look something like this:

```
# OCR disks
dasdo1:root:oinstall:0640
# Voting disks
dasdp1:oracle:oinstall:0640
# ASM disks
dasdq1:oracle:dba:0660
dasdr1:oracle:dba:0660
```

If using raw devices, the file should look something like this:

```
# OCR disks
raw/raw1:root:oinstall:0640
# Voting disks
raw/raw2:oracle:oinstall:0640
# ASM disks
raw/raw3:oracle:dba:0660
raw/raw4:oracle:dba:0660
```

The main reasons for creating a new file as opposed to modifying the existing file is that the Linux kernel loads the 49-oracle.permissions/51-oracle.permissions file before the 50-udev.permissions, ensuring that the correct file permissions get set. Another reason not to modify 50-udev.permissions is that the file can be overwritten with a support pack update. Without the correct file permissions, Oracle clusterware will not start correctly, causing issues with system startup.

The following command can be used to execute the assignment of the file permissions:

```
# /sbin/udevstart
```

You should then check the files to ensure that they have the correct file permissions set.

After the raw/block partitions are defined, you can then clear the disk being used for the OCR, Voting, and raw/block devices with the **dd** command:

```
dd if=/dev/zero of=/dev/dasdag1 bs=1K count=100000
```

At this point, the block/raw devices are ready for the installation of Oracle RAC using ASM. The Oracle Clusterware installation should be done first, followed by ASM disk group creation, and, finally, the Oracle RAC database creation.

4.6 Installing Oracle ASM with ASMLib

ASMLIB is an add-on utility for Oracle ASM that provides a method to assign disks to Oracle ASM disk groups. Oracle states that the purpose of ASMLib is to provide an alternative interface for ASM to identify and access block devices.

In order to use ASMLib, you first need to download and install three Linux RPMs based on the kernel version that you are using. You should first determine the level of the kernel that you are running with the following command:

```
[root@oralin01 ~]# uname -r
2.6.9-55.EL
```

The oracleasm-support 2.0.1 drivers are recommended, as they provide support for loopback, MD, LVM, and device-mapper devices.

You can download the RPMs from the following Oracle Web site:

http://www.oracle.com/technology/tech/linux/asmlib/index.html

And then run the following command:

```
# ls *.rpm
oracleasm-2.6.9-55.0.12.EL-2.0.3-1.s390x.rpm
oracleasmlib-2.0.2-1.s390x.rpm
oracleasm-support-2.0.3-1.s390x.rpm
[root@oralin02 tmp]# rpm -ivh oracleasm-support-2.0.3-1.s390x.rpm
Preparing...
########################################### [100%]
   1:oracleasm-support
########################################### [100%]
[root@oralin02 tmp]# rpm -ivh
oracleasm-2.6.9-55.0.12.EL-2.0.3-1.s390x.rpm
Preparing...
########################################### [100%]
   1:oracleasm-2.6.9-55.0.12
########################################### [100%]
[root@oralin02 tmp]# rpm -ivh oracleasmlib-2.0.2-1.s390x.rpm
Preparing...
########################################### [100%]
   1:oracleasmlib
########################################### [100%]
```

The next step is to configure the ASM Kernel. When we went to configure the ASM kernel, we initially received an error, as our Linux kernel version did not match exactly the kernel version of ASMLib kernel RPM:

```
[root@oralin02 tmp]# /etc/init.d/oracleasm configure
Configuring the Oracle ASM library driver.

This will configure the on-boot properties of the Oracle ASM library
driver.  The following questions will determine whether the driver is
loaded on boot and what permissions it will have.  The current values
will be shown in brackets ('[]').  Hitting <ENTER> without typing an
answer will keep that current value.  Ctrl-C will abort.

Default user to own the driver interface []: oracle
Default group to own the driver interface []: dba
Start Oracle ASM library driver on boot (y/n) [n]: y
Fix permissions of Oracle ASM disks on boot (y/n) [y]: y
```

```
Writing Oracle ASM library driver configuration: [  OK  ]
Creating /dev/oracleasm mount point: [  OK  ]
Loading module "oracleasm": Unable to load module "oracleasm"
[FAILED]
```

In order to correct this problem, we needed to refer to MetaLink Note: 462618.1 by running (based on the kernel version that you are running):

```
/usr/lib/oracleasm/oracleasm_debug_link 2.6.9-55.0.12.EL $(uname -r)
```

This creates a softlink in the Oracle ASM directory that can be verified with:

```
ls -l /lib/modules/$(uname -r)/kernel/drivers/addon/oracleasm
```

After this step, the ASM configuration worked. If at a later date you apply updates to the Linux kernel, you should follow this procedure again if a suitable driver is not available from the download site.

You should only do this workaround for minor revision differences between kernel drivers. For example, 2.6.9-55.0.12 to 2.6.9-55.0.13 and Oracle does not support this configuration. For production systems, you should ensure that your kernel version is the same as the ASMLib kernel versions that are available:

```
[root@oralin02 tmp]# /etc/init.d/oracleasm configure
Configuring the Oracle ASM library driver.

This will configure the on-boot properties of the Oracle ASM library
driver.  The following questions will determine whether the driver is
loaded on boot and what permissions it will have.  The current values
will be shown in brackets ('[]').  Hitting <ENTER> without typing an
answer will keep that current value.  Ctrl-C will abort.

Default user to own the driver interface [oracle]:
Default group to own the driver interface [dba]:
Start Oracle ASM library driver on boot (y/n) [y]:
Fix permissions of Oracle ASM disks on boot (y/n) [y]:
Writing Oracle ASM library driver configuration: [  OK  ]
Loading module "oracleasm": [  OK  ]
Mounting ASMlib driver filesystem: [  OK  ]
Scanning system for ASM disks: [  OK  ]
```

Once ASMLib is installed and configured, you need to mark the ASM disk with **createdisk** so that ASMLib can discover it You would do this as root under /etc/init.d.

If using Raw devices:

```
/etc/init.d/oracleasm createdisk DISK03 /dev/raw/raw3
/etc/init.d/oracleasm createdisk DISK04 /dev/raw/raw4
```

If using Block devices:

```
/etc/init.d/oracleasm createdisk DISK03 /dev/dasdq1
/etc/init.d/oracleasm createdisk DISK04 /dev/dasdr1
```

or

```
/etc/init.d/oracleasm createdisk DISK03 /dev/sda1
/etc/init.d/oracleasm createdisk DISK04 /dev/sdb1
```

In order for other RAC nodes to see the disk as well, you need to run **oracleasm scandisk** on each node so that each ASM instance can see the disk:

```
[root@oralin01 init.d]# ./oracleasm scandisks
Scanning system for ASM disks: [  OK  ]
```

You should then run /etc/init.d/oracleasm **listdisks** on each node to ensure that the results are the same on each node before proceeding with dbca and the Oracle ASM disk group creation.

4.7 Getting started with Oracle code

Once the infrastructure is set up, it is time to install the Oracle code:

1. Download files from OTN at http://otn.oracle.com.
2. Select **Database** → **10gR2 for zSeries Linux**. Download the following files:

   ```
   10202_zlinux_database.zip
   10202_zlinux_clusterware.zip
   ```

When you unzip the files, the high level directories are “database” and “clusterware”, respectively, and not “Disk1”.

The Cluster Verification Utility (CVU) should be run before CRS is installed. It can be found at **./clusterware/cluvfy/runcluvfy.sh**.

4.7.1 Oracle clusterware

Cluster Ready Services (CRS) is now called Oracle Clusterware. Oracle Clusterware should be installed in a separate Oracle Clusterware home directory.

This is a change to the Optimal Flexible Architecture (OFA) rules. You should not install Oracle Clusterware in a release-specific Oracle home mount point, typically /u01/app/oracle/product/10.2.0, as succeeding versions of Oracle Clusterware will overwrite the Oracle Clusterware installation in the same path.

If Oracle Clusterware 10g Release 2 (10.2) detects an existing Oracle Cluster Ready Services installation, then it overwrites the existing installation in the same path.

Oracle Clusterware manages the shared disk integrity, enabling two or more Linux guests to write to the same shared disk, much like VM manages shared mini-disks

4.7.2 Cluster Verification Utility

Oracle Database 10g Release 2 (10.2) is provided with the Cluster Verification Utility. Cluster Verification Utility (CVU) is a validation tool that you can use to check whether or not your cluster is properly configured, to avoid installation failures, and to avoid database creation failures.

The CVU is used in the configuration and installation of Oracle Clusterware and RAC. The command **runcluvfy** will verify all the important components necessary for a RAC installation. This must run (stage -pre) successfully before installing the clusterware.

The **crs_stat** command provides view of resources controlled by CRS. The -t option gives a short listing of the contents of the cluster registry

The **checkcrs** command checks the cluster components, such as

- Global services daemon (GSD)
- Oracle Notification Service (ONS)
- Instances
- Listeners
- Cluster Virtual IP addresses (VIP)

If you have completed the setup of the network and the disks correctly, this utility should run successfully.

4.7.3 Note on LVM

Oracle's statement that they do not support LVM for RAC does not apply to Linux for System z.

Oracle does not support LVM with Oracle RAC because it does not have clustering capabilities. This has led to some confusion regarding IBM System z, as we use LVM not for clustering but to create one large logical volume if we are using ECKD devices.

On IBM System z, you can use LVM with Oracle Single Instance or RAC if you want to create one large logical volume from several small ECKD devices. For example, if you have two 7 GB drives and you need an 8 GB drive to install Oracle Clusterware and Binaries on each node, you can use LVM to create one larger logical volume.

Note: In several cases we noticed that the CPU utilization was 20% to 30% with just CRS running, before the database was started. We applied patch 6056783 and this resulted in the CPU utilization going down to about 2%. If you are running 10gR2, you may want to investigate using this patch.

4.8 More information about Oracle RAC

The details of the complete installation process has already been documented in several IBM Redbooks publications and white papers so the content is not duplicated here. This chapter highlights the environment setup.

For more information about Oracle for System z, refer to these books, which can be found at http://www.redbooks.ibm.com:

- *Experiences with Oracle 10g Database for Linux on zSeries*, SG24-6482
- *Experiences with Oracle?® 10gR2 Solutions on Linux for IBM System z*, SG24-7191

  This covers a RAC on Red Hat 4 installation.

Also, see the following white paper, which can be found at http://www.oracleracsig.org:

Building an Oracle RAC 10g R2 cluster for Linux for System z

5

Getting started using z/VM for Oracle Database 10gR2

To obtain optimum performance from Oracle RDBMS when running in Linux for System z, the proper configuration of both Linux and z/VM is extremely important. Our experiences have shown that the vast majority of performance concerns were solved by making changes to z/VM, Linux, or the I/O subsystem. This chapter discusses the major areas of concern.

- Sizing and tuning memory
- CPU allocation
- Paging, swap, and I/O considerations
- Monitoring performance

For more information about VM tuning recommendations, refer to:

http://www.vm.ibm.com/perf/tips/

This chapter was previously published in *Experiences with Oracle 10g Database for Linux on zSeries*, SG24-6482. We have updated it with current information.

5.1 Memory sizing and configuration

The sizing and allocation of memory is one of the most critical areas of implementing Oracle in Linux under z/VM. This was especially so with 31-bit Oracle9i and 31-bit Linux distributions. Even though Oracle Database 10g 64-bit runs in 64-bit Linux, memory sizing still affects performance.

When running under z/VM, there is a need for expanded storage. Even though z/VM is fully 64-bit and supports 64-bit virtual machines, expanded storage must be configured when running Oracle. You should configure 10% of the memory in the LPAR as expanded storage, not to exceed 2 GB. Once configured, it is unlikely that the amount of expanded storage configured will change.

When sizing the Linux virtual machine for Oracle, the virtual machine should be configured with the minimum amount of memory needed. Linux will set up cache buffers with all memory allocated to it. It is better to let z/VM handle paging and manage memory where possible. System performance can be adversely affected by the amount of memory allocated.

The method to size a guest for Oracle is to add the System Global Area, the Program Global Area (if needed), and about 128 MB for Linux. If you are using ASM, another 256 MB should be added for it. If this is a large database, this amount of memory *may need to be larger*. Check the *Oracle ASM Administration Guide* for instructions to size ASM SGA memory requirements.

The System Global Area is the memory requirement for Oracle. This is shared memory within Oracle to functions such as caching table data, parsing SQl, sorting, and so on.

The Program Global Area is memory outside the Oracle instance. This generally is used by users who create connections to the database. Depending on the application, this can be either a small amount of memory per user or very large amounts (in excess of 10 MB per user) for applications like the Oracle E-Business Suite.

This situation tends to be the complete opposite of assigning memory to Linux in an Intel environment or even in UNIX®. But it must be remembered that in this case, Linux is running under and being managed by the z/VM Hypervisor.

5.2 CPU allocation

CPU (or CP) allocation can be viewed from several perspectives:

- A MIP's requirement for the databases and applications to be run in Linux for System z
- Allocation of virtual CPs
- Setting shares or running multiple LPARS with CPUs either as shared or dedicated

5.2.1 Sizing

As part of the consolidation process, workloads should be sized not only to ensure that the correct amount of MIPS are determined, but also to aid in determining if this is a good candidate to move to Linux for System z. It can also help determine if this database should possibly be run in a Linux virtual machine or in an LPAR. The other databases, such as development, test, and so on, should be considered as candidates to be run under z/VM. This is a very important part of the consolidation process, if not the most important part.

Doing the right work here will save problems later after the workload is moved to Linux for System z.

IBM can provide sizing estimates, and there is no charge for this. To do an estimate, IBM will need to following information:

- Make and model of the system on which the database currently runs
- The number of CPs and the MHz rating of the CPs
- The peak utilization of each system

It is best if these are actual numbers from a system tool such as **vmstat**. Approximations can be provided, but the amount of utilization the estimates are in error is directly proportional to the error in the sizing.

There are several methods of sizing. IBM has several tools it uses based on information gathered from customers. Customers can also size servers and create TCO values by purchasing a license for the zRACE tool from Ideas International (http://www.ideasinternational.com).

5.2.2 CP allocation

The amount of virtual CPs allocated to a Linux machine is important. In general, one virtual CP is a good starting point unless the sizing points to something larger.

In a constrained CPU environment (that is, more CP cycles are needed to complete the work), you should give the Linux virtual machine all the CPs it needs to do its work efficiently. However, there should not be more CPs allocated than the number of physical CPUs that are assigned to this LPAR when running under z/VM. If this were to happen (more virtual CPs than physical CPUs), performance would suffer as the cost to execute the workload would increase. This is due to an increased scheduling impact in z/VM's control program.

CPU time is limited to the number of physical CPUs installed. Therefore, it would be prudent to reduce CPU usage where possible. This would be in areas such as:

- Eliminate any unnecessary services that might install with the Linux guest.
- Eliminate any unnecessary cron tasks.
- Reduce unnecessary work.
- Eliminate using “r-u-there” pings to determine if the virtual machines are there and up.
- Do not measure idle guests. Measuring takes cycles.
- Network consideration.
- Use VSwitch.

5.2.3 Setting share values

The workload of Linux virtual machines can be managed or controlled by setting the SHARE value of a virtual machine. The first choice of SHARE is to use ABSOLUTE or RELATIVE. The second choice is the size of the SHARE. The simplest way to decide which to use for a specific server is to determine the following: as more users log on to this system, should this service machine get more CPU or less?

A relative share says this server should get a relative share of the processor, relative to all virtual machines in the dispatch and eligible lists. As more users log on, the share will drop.

An absolute share remains fixed up to the point where the sum of the absolute shares is 100 percent or more, which is a rather confused state of configuration. Servers might have a requirement that increases as the level of work increases.

These servers should have ABSOLUTE shares. All other users should use RELATIVE.

The size of the share is both a business decision and a performance decision. For example, if one server is assigned a very high share, which might be as much of the system as the rest combined, one would expect this server to be absolutely critical to your business. This server has the capability of taking resources whenever it needs them, but if this server starts looping, it would easily consume all the resources allocated. But in general, this is not a likely situation for production.

A simple way of looking at share values is that if there is a heavy contention for the processor, what servers would you like to run? The production Oracle databases are an obvious choice. Required servers should have absolute values.

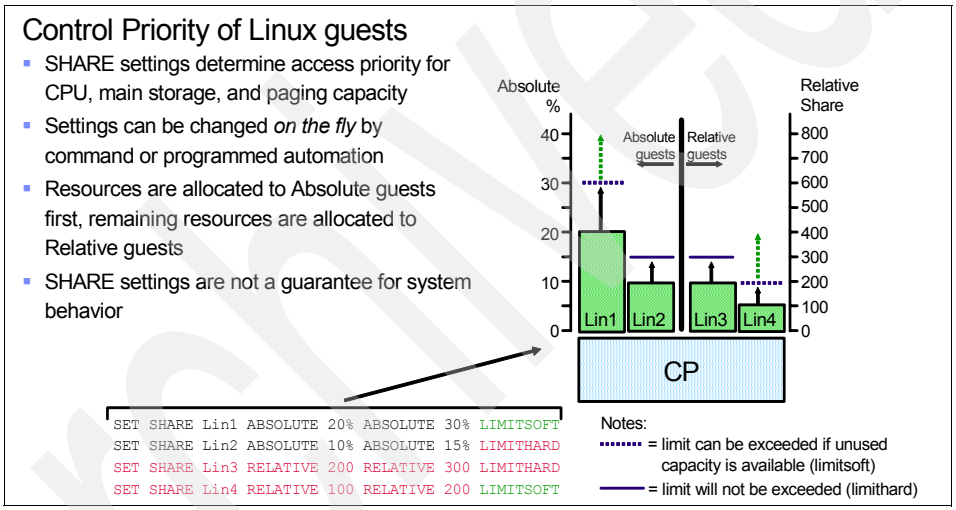


Figure 5-1 Using set share commands to manage guest priorities

5.3 Paging, swapping, and I/O considerations

z/VM uses paging, while Linux swaps. Swapping in Linux is really paging. Swapping is a “carry-over” term from the early days of Linux. So Linux does page, using its own paging algorithm.

The approach to this is to keep the memory in Linux virtual machines small enough so that z/VM does not do a lot of paging and large enough so that Linux does not have to do excessive paging. And all paging at the first level should be to a memory device, such as a VDISK for Linux or expanded storage for z/VM.

5.3.1 z/VM paging

One of the common mistakes with new VM customers is to not allocate paging space (along with not configuring expanded storage). The installation process configures enough paging space to complete an installation. This paging space on the sysres pack is small and can handle a small amount of tasks. However, you should remove the paging space from the sysres pack and add DASD page space to do real work. The *VM Planning and Administration* manual has details on determining how much space is required. Here are a few thoughts:

- If the system is not paging, you may not care where you put the page space. However, it has been our experience that sooner or later the system grows to a point where it pages and then you might wish you had thought about it.
- z/VM paging is most optimal when it has large contiguous available space on volumes that are dedicated to paging. Therefore, do not mix page space with other space (user, tdisk, spool, Linux guests, and so on).
- Set up VM paging to many disks. The more subchannels z/VM can use, the faster the paging will take place. It is best to use smaller 3390s, such as the mod 3s or mod 9s.
- A rough starting point for page allocation is to add up the virtual machine sizes of virtual servers running and multiply by 2. Keep an eye on the allocation percentage and the block read set size.

5.3.2 Swap space for Linux

Try to avoid swapping in Linux whenever possible. However, sometimes swapping is unavoidable. There are a few acceptable ways to set up swap space:

- Traditional minidisk: Using a traditional minidisk on physical DASD requires some setup and formatting the first time and whenever changes in size of swap space are required. However, the storage burden on z/VM to support minidisk I/O is small, the controllers are well-cached, and I/O performance is generally very good. If you use a traditional minidisk, you should disable z/VM Minidisk Cache (MDC) for that minidisk (use the MINIOPT NOMDC statement in the user directory).
- VM VDISK: A VM virtual disk in storage (VDISK).

VDISK is backed by a memory address space, not by real DASD. While in use, VDISK blocks reside in main storage (which make them very fast). When not in use, VDISK blocks can be paged out to expanded storage or paging DASD.

Linux assigns priorities to swap extents. You could set up a small VDISK with higher priority (higher numeric value), and it would be used for swap as long as there was space on the VDISK. Swap extents of equal priority are used in

round-robin fashion Setting up two VDISKS of different priorities or a VIDSK and a minidisk with the higher priority on the VDISK can provide a tuning tool or method for properly sizing memory needs on the Linux virtual machine. Monitor the paging at peak times; if there is paging to the second paging disk, then you should consider adding virtual memory to Linux.

5.3.3 I/O considerations

One of the biggest performance issues we ran into during some testing, both on z/OS® and Linux for System z, was when we placed the database files on a single rank in the IBM Storage. This will cause many performance-related issues that mostly involve cache utilization. The best practice for the DS or ESS system is to distribute data across as many ranks (arrays) as possible. This made a dramatic difference in our testing, even though this was a small database. It is also important to stripe the logical volumes that are created. This enables the operating system to issue multiple I/Os against the LVM. Striping a logical volume is not the same as any striping or RAID that is done inside the disk controller. The guidelines for striping on a IBM Total Storage ESS and DS8000™ are as follows;

- ESCON: One stripe per channel
- FICON: One stripe per physical volume
- FCP: One stripe per volume or LUN

When the logical volume is created, one of the parameters is the stripe size. The best stripe size is dependent on the workload characteristics. The IBM Web site recommends 16 KB or 32 KB stripe sizes.

If you are using Oracle ASM, then ASM will stripe the volume it creates.

A final thought on this is that faster and newer is always better. Some implementations have experienced up to a 50 percent performance improvement just going from ESCON to FICON.

5.3.4 Monitoring the system

z/VM provides extensive virtualization techniques. A virtual machine running under z/VM acts as though it has all the resources allocated to it in the user direct profile that was created for the virtual machine. This means that monitoring what a virtual machine is doing by looking at resource utilization with a tool or monitor running in that virtual machine may not always be accurate.

Since monitoring a virtual machine from the virtual machine is not always accurate, it is extremely important to use a tool to monitor the z/VM system and all the virtual machines from a system perspective. There are two tools that we have experience with and both provide value:

- IBM Performance Toolkit for VM—Licensed through IBM
- ESAMON from Velocity Software (http://www.velocity-software.com)

While using one of these tools is absolutely necessary for performance monitoring and tuning, do not overlook using tools like **sar** to **vmstat** monitor events and resource utilization, such as swapping in a Linux virtual machine. Yet another way to monitor performance, especially during times when problems are suspected, is the use of CP commands such as **cp** indicate load. These are very valid to use for a current point in time, but do not provide the ability to go back to review past performance, as does one of the tools mentioned above.

This chapter described the configuration of Linux and z/VM. However, this is for an Oracle database, and using a tool such as Statspack (or DB Control AWR) is still an important way to monitor and tune the Oracle database server.

5.4 Summary

Here is a summary of the key points needed to get the best performance from Oracle:

- Memory: This is likely the most critical area of tuning Linux virtual machines. The mindset here is the exact opposite of Linux on Intel: the less memory, the better. Memory assigned to Linux should be enough to cover the needs of the Oracle database (SGA + PGA) and about 128 MBs more for Linux. Add 256 MBs for the ASM SGA if ASM is being used.
- Monitor resources: The more virtual machines installed under z/VM, the more you will need to use a system monitor, such as the z/VM Toolkit. Trying to understand performance problems and determine capacity needs without this type of monitor is impossible.
- Paging and swap space: Paging at the z/VM level and Linux level (swapping) is unavoidable. You must allow for this situation. Ensure that the z/VM paging space is less than 50% utilized during peaks. The best paging device for Linux is a VDISK.
- Avoid I/O bottlenecks: Use best practices for configuring and using disk. Data must be distributed as evenly as possible across different ranks. The logical volumes should be striped, with the number of stripes being dependent on the technology used.

6

Oracle Business Intelligence Enterprise Edition with the database on Linux for System z

Oracle Business Intelligence Enterprise Edition (BIEE) is an application that can be run on a supported middle tier with the database that contains the user data being hosted on Linux for System z. This chapter describes the steps we followed to install this architecture using Linux Intel as the middle tier and Oracle 10gR2 on Linux for System z as the DB tier.

Installing the Database code and the Application Server code is covered in detail in *Experiences with Oracle 10g Database for Linux on zSeries*, SG24-6482. This chapter concentrates on the new BIEE component.

6.1 Overview

This chapter describes the successful installation and setup of Oracle's Business Intelligence Enterprise Edition, the BIEE suite of products. This configuration includes the components list in Table 6-1.

Table 6-1 BIEE components

Component	Platform
Oracle 10gR2 Database	IBM System z VM 5.1 and Linux
Oracle 10gR2 Database Client	x86 SUSE SLES 9
Oracle Application Server V0.1.3.1 Oracle Application Server patch V10.1.3.3.3	x86 SUSE SLES9
Oracle BIEE V10.1.3.2.	x86 SUSES SLES 9
Oracle BIEE V10.1.3.2 Administration	x86 Windows 2003 Server SP2

Figure 6-1 on page 97 describes the configuration of components as they were installed for this exercise. The z990 was located in Salt Lake City at Oracle's Siebel Data Center and the System x™ resources were located in California at Oracle's Headquarters facility in Redwood Shores.

The steps in the installation process are:

- On System z platform
  - Install the Oracle Database code and create an Oracle database.
- On x86 Linux platform
  - Install AS10g - 10.1.3.1.0 and patch 10.1.3.3.0.
  - Install BIEE - 10.1.3.3.2.
- On x86 Windows Platform
  - Set up the BIEE Administrator.
- Connecting the x86 and Linux for System z platform
  - Configure the BIEE platform to connect to the database on the System z server.

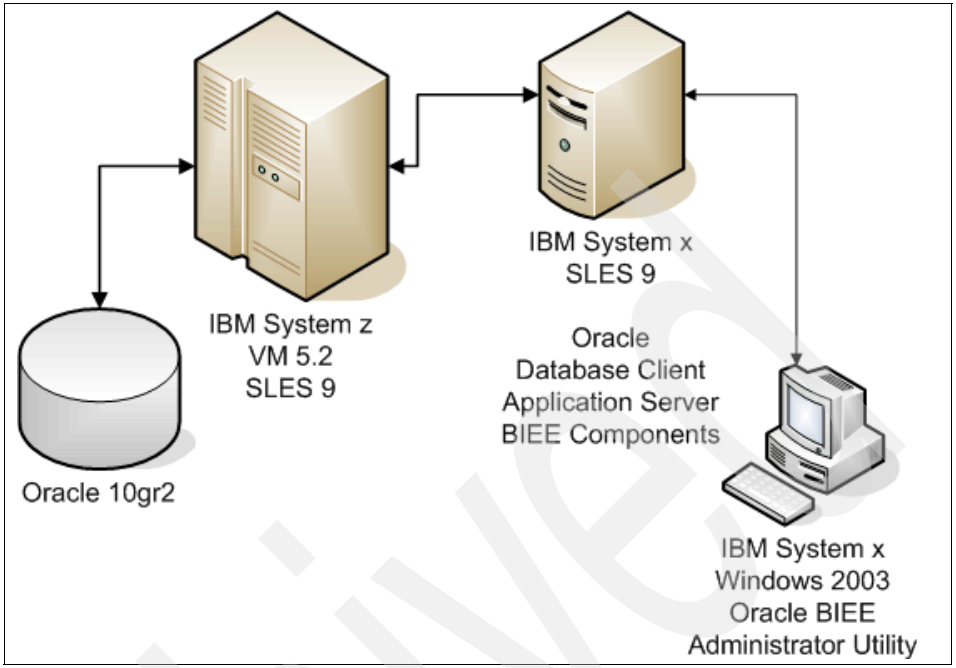


Figure 6-1 Oracle BIEE configuration

This chapter is written as a supplement to the following Oracle Documentation from the Oracle BIEE Documentation Library (E10415_01.zip):

- *Oracle Business Intelligence Infrastructure Installation and Configuration Guide*, B31765-03
- *Oracle Business Intelligence Infrastructure Quick Installation Guide*, B31764-01
- *System Requirements and Supported Platforms for Oracle Business Intelligence Suite* Enterprise Edition, E10417-02
- *Oracle Application Server Installation Guide 10g Release 3 (10.1.3.1.0) for Linux x86*, B31013-01
- *Oracle Database Client Installation Guide 10g Release 2 (10.2) for Linux x86*, B15662-02

6.2 Install System z components

This installation starts with the database code and database creation on a System z 990.

This chapter does not include the detailed installation process of the database component, because that process is described in Chapter 3, “Installing an Oracle 10gR2 single instance” on page 19.

Once the Oracle Database Code was installed, the database was created and populated with SQL command files and data provided by Oracle for the BIEE testing.

The database was created using an SQL script create_db.sql, a portion of which is shown in Example 6-1. The results of running the script are shown in Example 6-2 on page 99.

The following commands were used to execute it:

```
cd /home/oracle
SQLPlus
SQL>
SQL>create_db.SQL
```

Example 6-1 create_db SQL found in /home/oracle — excerpt

```
set echo on
set feedback on

connect / as sysdba
shutdown abort
spool crdb_dbolap.log
connect / as sysdba
startup nomount

create database dbolap
   controlfile reuse
   maxinstances 2
   maxlogfiles 12
   maxdatafiles 100
   noarchivelog
   character set AL32UTF8
   datafile
      '/oracle/dbf/dbolap_system.dbf' size 8000M REUSE
      extent management local
   sysaux datafile
```

```
      '/oracle/dbf/dbolap_sysaux.dbf' size 2000M REUSE
   logfile
      '/oracle/dbf/dbolap_log1.dbf' size 500M REUSE,
     .,
      '/oracle/dbf/dbolap_log4.dbf' size 500M REUSE
   DEFAULT TABLESPACE DATA DATAFILE
      '/oracle/dbf/dbolap_data1.dbf'  size 15000M REUSE,
      .
      .
      .
     '/oracle/dbf/dbolap_data20.dbf' size 15000M REUSE
      extent management local autoallocate segment space management
auto
   default temporary tablespace TEMP tempfile
      '/oracle/dbf/dbolap_temp1.dbf' size 10000M REUSE
   undo tablespace UNDO datafile
      '/oracle/dbf/dbolap_undo1.dbf' size 20000M REUSE
/

connect / as sysdba
set echo off
set feedback on
@/oracle/rdbms/admin/catalog.sql
@/oracle/rdbms/admin/catproc.sql
@/oracle/rdbms/admin/catexp7.sql
@/oracle/rdbms/admin/catexp.sql
@/oracle/rdbms/admin/catdbsyn.sql
@/oracle/sqlplus/admin/pupbld.sql
#MANY LINES NOT INCLUDED IN THIS EXAMPLE
```

Example 6-2 Execution output from running the create_db script

```
SQL> connect / as sysdba
Connected to an idle instance.
SQL> startup nomount
ORACLE instance started.

Total System Global Area 2147483648 bytes
Fixed Size                2072504 bytes
Variable Size           486539336 bytes
Database Buffers       1627389952 bytes
Redo Buffers             31481856 bytes
SQL>
SQL> create database olapmst
  2     controlfile reuse
```

```
  3     maxinstances 2
  4     maxlogfiles 12
  5     maxdatafiles 100
  6     noarchivelog
  7     character set AL32UTF8
  8     datafile
.
.
.
.
SQL>
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area 2147483648 bytes
Fixed Size                  2072504 bytes
Variable Size             486539336 bytes
Database Buffers         1627389952 bytes
Redo Buffers               31481856 bytes
Database mounted.
Database opened.
```

6.2.1 Populating the database

A export file was provided to us to populate the database. The database was populated using the following import command.

```
imp oraperf/oraperf full=y file=oraperf.dmp buffer=2048000 commit=y
log=oraperf.imp.log
```

After the import ran, the log showed that the data was being inserted successfully, as shown in Example 6-3.

Example 6-3 Log of import execution

```
Connected to: Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - 64bit
Production
With the Partitioning, OLAP and Data Mining options

Export file created by EXPORT:V10.01.00 via conventional path
import done in US7ASCII character set and AL16UTF16 NCHAR character set
```

```
import server uses AL32UTF8 character set (possible charset conversion)
export client uses AL32UTF8 character set (possible charset conversion)
. importing ORAPERF's objects into ORAPERF
. . importing table                  "CATEGORIES"          8 rows imported
. . importing table                "CATEGORIES_L"       9800 rows imported
. . importing table                "CATEGORIES_M"        119 rows imported
. . importing table                   "CUSTOMERS"         21 rows imported
. . importing table                 "CUSTOMERS_L"      29400 rows imported
. . importing table                 "CUSTOMERS_M"        315 rows imported
.
.
.
. . importing table      "sales_supplier_customer"        994 rows imported
. . importing table    "sales_supplier_customer_l"          0 rows imported
. . importing table    "sales_supplier_customer_m"          0 rows imported
. . importing table                 "salesplan_l"          0 rows imported
. . importing table                 "salesplan_m"          0 rows imported
. . importing table                  "shippers_m"          0 rows imported
. . importing table                     "sites_m"          0 rows imported
. . importing table                 "suppliers_m"          0 rows imported
About to enable constraints...
Import terminated successfully without warnings.
```

6.3 Install x86 components

System preparation for the installation of the database client, application server, and BIEE products included installing SLES 9, SLES 9 Service Pack 3, and the additional packages called for by the individual products. The packages for the Oracle database client are listed in Example 6-4. The additional packages for the Oracle Application Server and its patch are listed in Example 6-5 on page 102.

Example 6-4 Oracle Database Client software requirements

```
Binutils-2.15.90.0.1.1-32.5
Gcc-3.3.3-43.24
Gcc-c++-3.3.3-43.24
glibc-2.3.3-98.28
gnome-libs-1.4.1.7-671.1
libstdc++-3.3.3-43.24
libstdc++-devel-3.3.3-43.24
make-3.80-184.1
pdksh-5.2.14-780.1
sysstat-5.0.1-35.1
```

```
xscreensaver-4.16-2.6
```

Example 6-5 Oracle Database Server software requirements

```
glibc-2.3.3-98.28
Gcc-3.3.3-43.24
gcc-c++-3.3.3-43.24
libstdc++-3.3.3-43.24
libstdc++-devel-3.3.3-43.24
openmotif21-libs-2.1.30MLI4-119.1
pdksh-5.2.14-780.1
make-3.80-184.1
gnome-libs-1.4.1.7-671.1
gnome-libs-devel-1.4.1.7-671.1
sysstat-5.0.1-35.1
binutils-2.15.90.0.1.1-32.5
```

Once the database prerequisites were installed, Oracle Database Client V10.2.2.1 was installed. The Oracle Application Server was installed next. The Oracle Application Server installation took place in two parts:

- The base OAS 10.1.3.1.0
- A patchset, which was then patched to 10.1.3.3

Table 6-2 lists the components to download as of January 2008.

Table 6-2 Installation components

Component name	File name	File size	Installation command
10gR2 Client	10gR2 Clientx86.zip	478351419	runInstaller
OAS v 10.1.3.1.0	soa_linux_x86_101310_disk1.cpio	670502400	runInstaller
OAS v 10.1.3.3.0	o6148874_101330_LINUX.zip	552482303	runInstaller
BIEE v 10.1.3.3.2	biee_linux_x86_SUSE_101332_disk1.cpio	1514908672	setup.sh

6.3.1 Preparing the environment

Use YAST2 to ensure that the unique server software requirements for BIEE v10.1.3.3.1 are satisfied. The GNU Compiler Collection (GCC) C++ runtime Version 3.4.3 (or later) libraries must be installed prior to installing BIEE. These libraries are listed in YAST2 as:

```
compat.libstc++-lsb  The standard C++ Shared Library
```

This project assumes that the 10gr2 database client has been installed on the i386 host on which the BIEE servers are installed.

Ensure that the oracle SID is in your .profile statement and that the TWO_TASK variable has been set to reflect the proper Oracle SID. Putting these in your .profile prevents the mistake of forgetting to set the SID:

```
/* Excerpt from .profile */
   export ORACLE_SID=olapmst
   export TWO_TASK=$ORACLE_SID
```

The installation library used was FTPd to the middle tier and expanded with the following command:

```
cpio -idmv < biee_linux_x86_SUSE_101332_disk1.cpio
```

This created the directory /oracle/SS_Linux/Server/Oracle_Business_Intelligence.

The installation process was started using the following commands:

```
cd /oracle/SS_Linux/Server/Oracle_Business_Intelligence
./setup.sh
```

If you get an error message that you are missing the gcc library, you will need to return to 6.3.1, “Preparing the environment” on page 102 and re-install the necessary GCC C++ runtime libraries.

When the setup is correct, the installation process will display Figure 6-2. This first splash window is a space advisory for the Installation. We found that we used 2.1 GB during our installation.

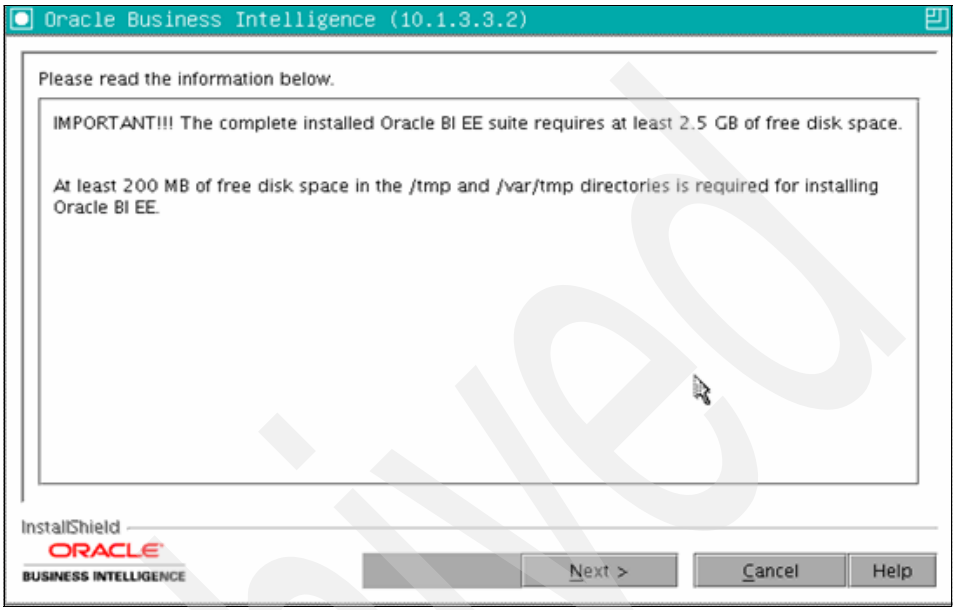


Figure 6-2 First window in BIEE installation

We installed the BIEE code in the /oracle/BIEE directory, as shown in Figure 6-3 on page 105.

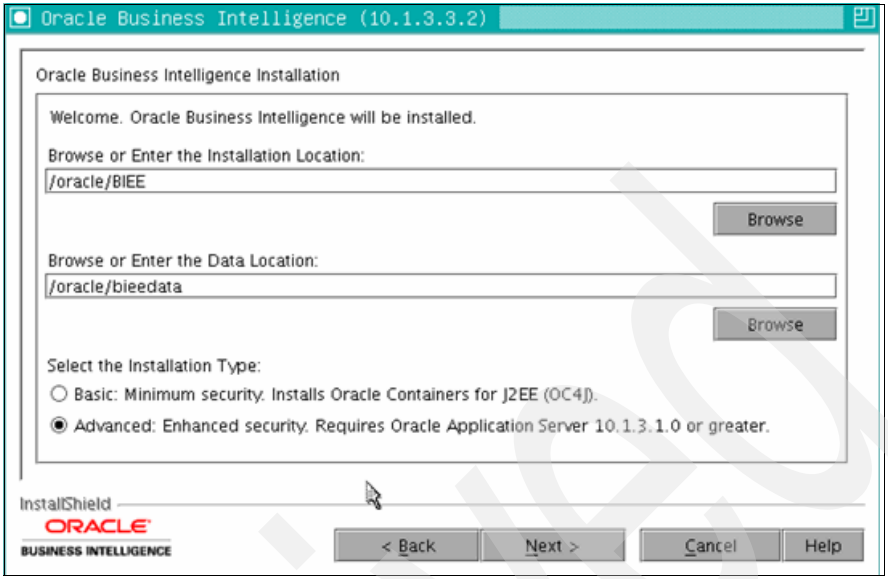


Figure 6-3 BIEE directory selection for installation

The selection of the type of installation is a site specific call. This project used the complete selection, as shown in Figure 6-4, because the Oracle recommendation in the installation manual recommends the complete install.

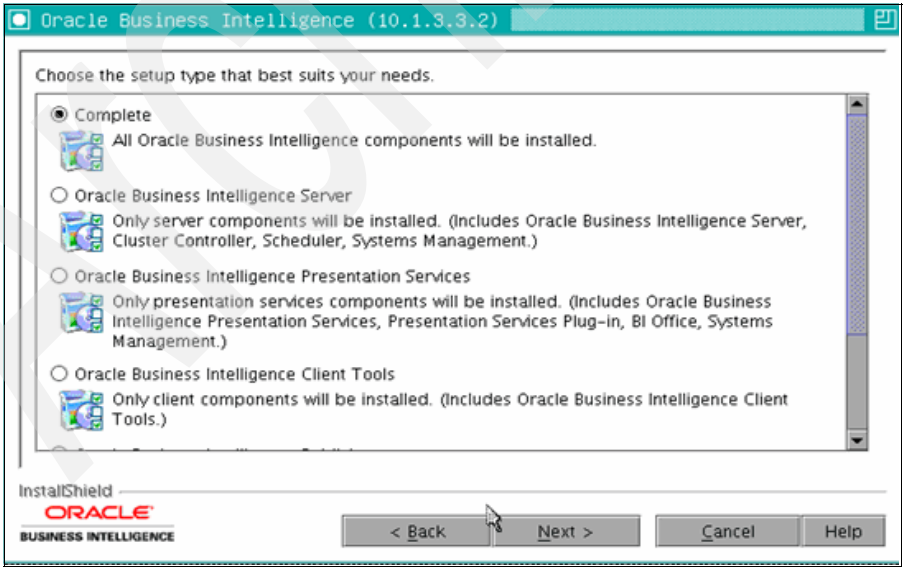


Figure 6-4 Oracle BIEE setup type

The Oracle Application Server selection window provides the linkage to between the BIEE product suite and the Oracle Application Server, as shown in Figure 6-5.

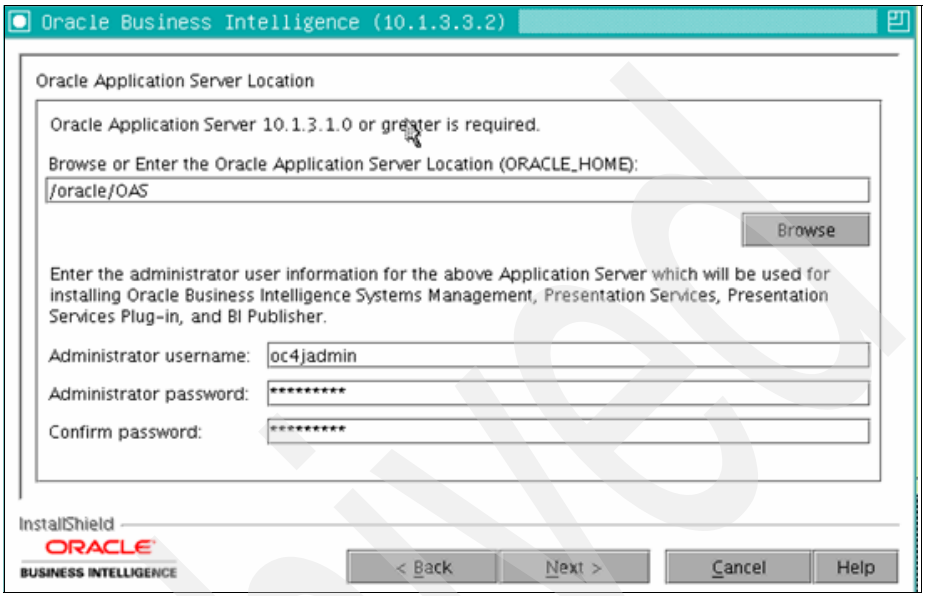


Figure 6-5 BIEE OAS location

In this case, we selected English as the language to use to display any error messages from the BIEE product, as shown in Figure 6-6 on page 107.

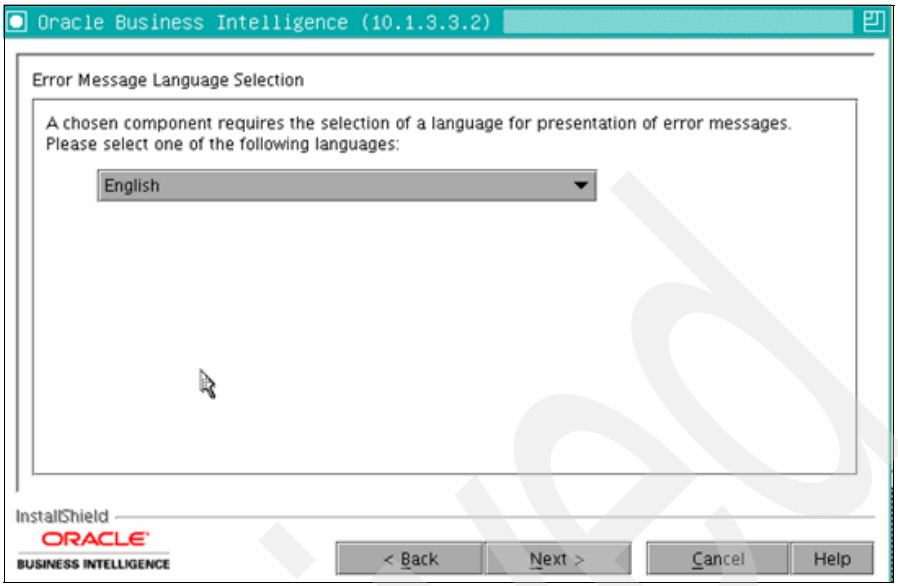


Figure 6-6 Oracle BIEE language selection

The Summary window is displayed in Figure 6-7. Select **Next** and the installation will commence, as shown in Figure 6-8 on page 108. Our project completed the actual infrastructure installation in less than 30 minutes.

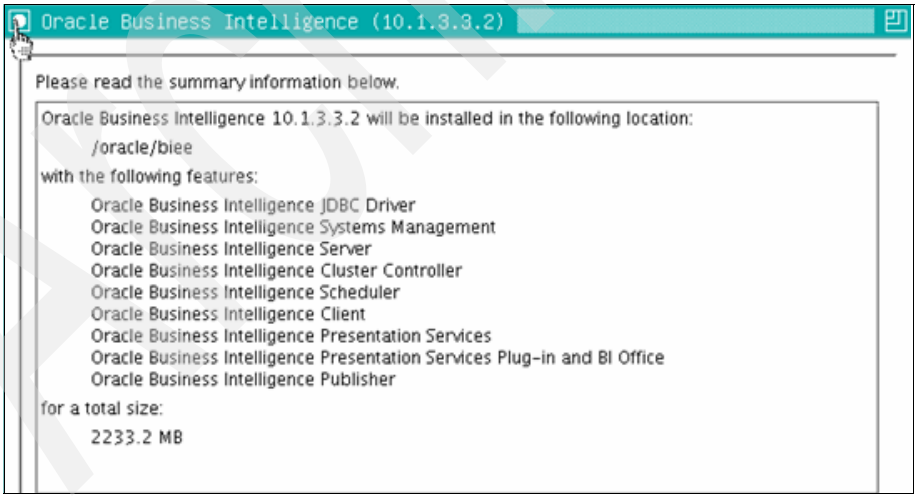


Figure 6-7 Oracle BIEE product summary

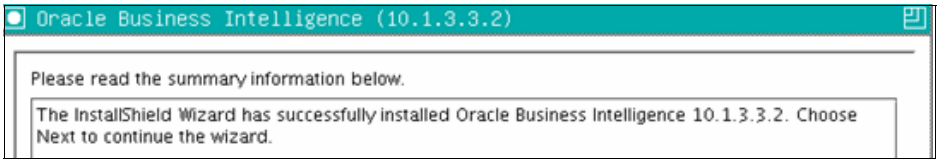


Figure 6-8 Oracle BIEE installation

The installation of BIEE 10.1.3.3.2 server infrastructure is now complete, as shown in Figure 6-9.

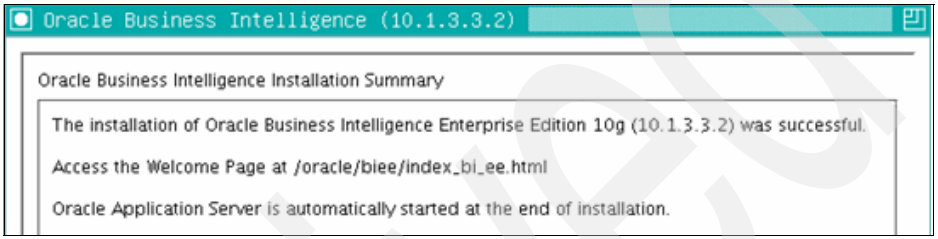


Figure 6-9 Oracle BIEE installation completion

6.4 OAS startup and BIEE welcome page

Figure 6-10 on page 109 shows the welcome window for BIEE.

Figure 6-10 Welcome window for BIEE

The BIEE Infrastructure is now complete. The welcome window can be displayed. It contains management links for the components that have been installed. The installation will automatically initialize the BIEE components on Linux, including:

- Oracle Business Intelligence Server process
- Oracle Business Intelligence Presentation Services process
- Oracle Business Intelligence Javahost process
- Oracle Business Intelligence Scheduler process

6.4.1 Post-installation tests

The server can be tested using the **run-sa** command, as shown in Example 6-6.

Example 6-6 Testing the server

```
oracle@I78OX009:/oracle/biee/setup> ./run-sa.sh start
Oracle BI Server startup initiated.
Please wait for a while for the Oracle BI Server to completely start.
Execute the following command to check the Oracle BI Server logfile and see if it
started.
    tail -f /oracle/biee/server/Log/NQServer.log
oracle@I78OX009:/oracle/biee/setup>
oracle@I78OX009:/oracle/biee/setup>  tail -f /oracle/biee/server/Log/NQServer.log
2008-01-22 15:14:47
```

```
     [36007] Loading repository /oracle/biee/server/Repository/paint.rpd.
2008-01-22 15:14:47
     [14055] Loading subject area: Paint ...
2008-01-22 15:14:47
     [14056] Finished loading subject area: Paint.
2008-01-22 15:14:47
     [58002] Query Cache loaded with 0 entries from saved cache files.
2008-01-22 15:14:47
     [43030] :  Oracle BI Server started.  Version: 10.1.3.3.2.071217.1900.
```

At this point, you can install the individual Oracle BI components (please see the details in the *Oracle Business Intelligence Infrastructure Installation and Configuration*, Guide, B31765-03). These components include:

- Disconnected Client
- Briefing Book Reader
- Oracle BI Office Plug-In
- Oracle BI Open Intelligence Interface
- Oracle BI Publisher Desktop

6.5 Connecting the BIEE tier to the DB tier

The Oracle database definition is accomplished using the Oracle BIEE Administration tool. This is installed on a Windows host and is used to administer the BIEE environment. The Administration Tool is part of the Windows deployment for Oracle BIEE. The file name is Biee_windows_x86_101332_disk.zip (1464447694 bytes).

This is expanded to three directories on a Windows server machine. This projects setup was invoked as follows:

```
C:\FTP\BIEE Client
Install\Server\Oracle_Business_Intelligence\setup.exe
```

This command invokes the installation process, which starts with an advisory for space, as shown in Figure 6-11.

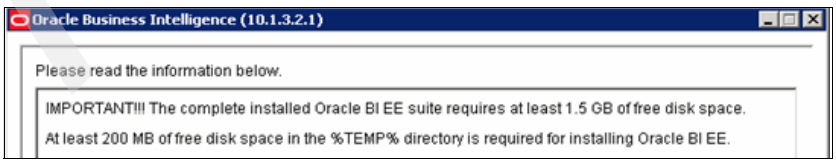


Figure 6-11 Space message

The product and data location selection is next in the process (see Figure 6-12). These are site specific.

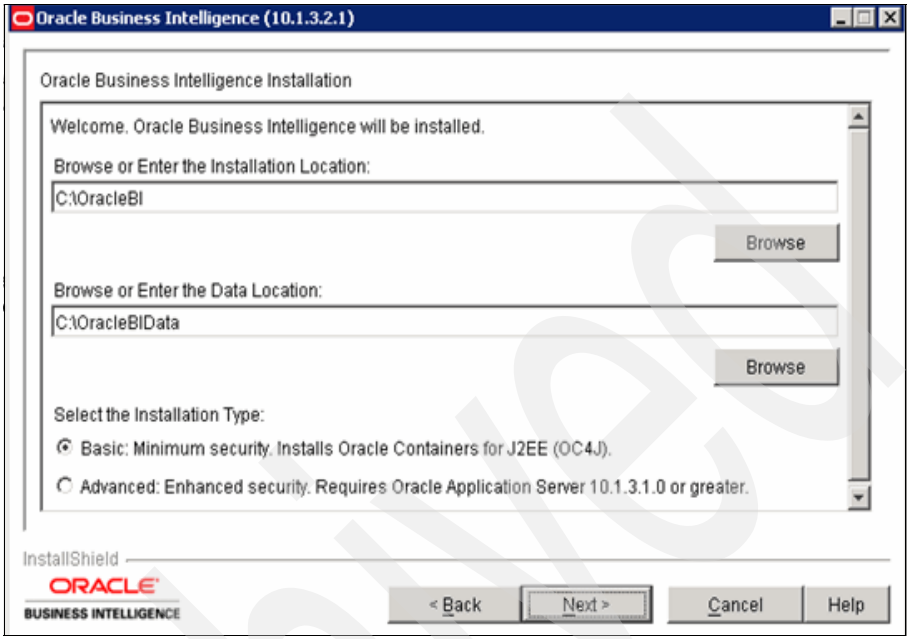


Figure 6-12 Location choices

Oracle BIEE requires Java JDK™ 1.5 or above and the knowledge of its location (see Figure 6-13).

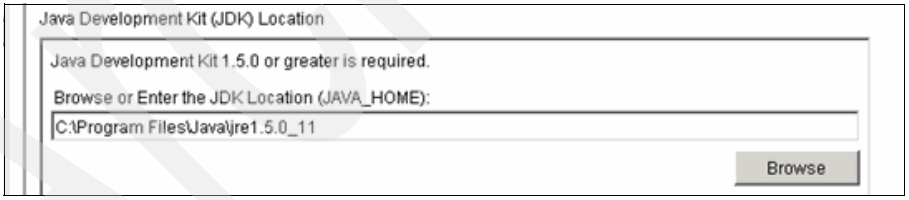


Figure 6-13 JDK location

This project only installed the Oracle BIEE Administration tools; to minimize the impact on the Windows machine, we selected **Custom**, as shown in Figure 6-14.

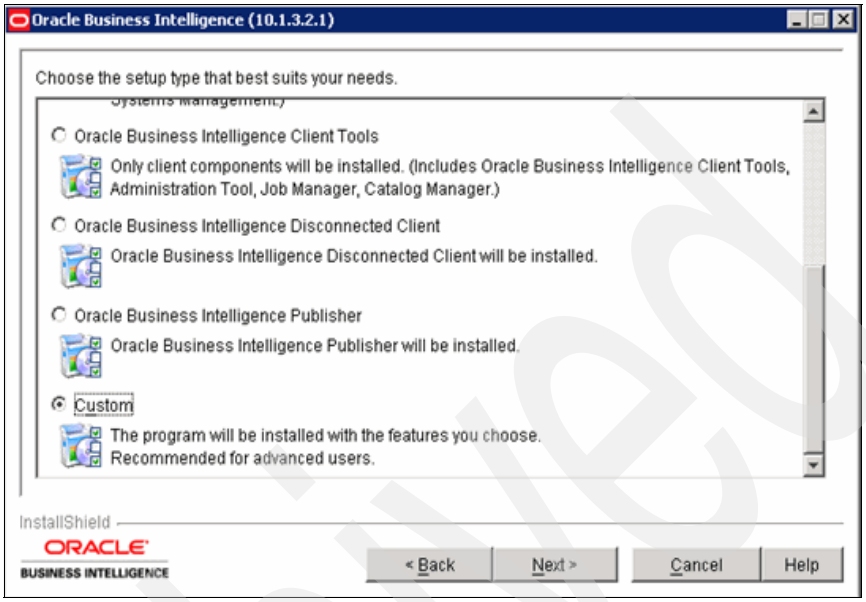


Figure 6-14 Setup choice

In Figure 6-15 on page 113, a custom selection offers the installer the choice of what components to install. For this project, we selected the following:

- ODBC Driver
- Administration Tool
- Client

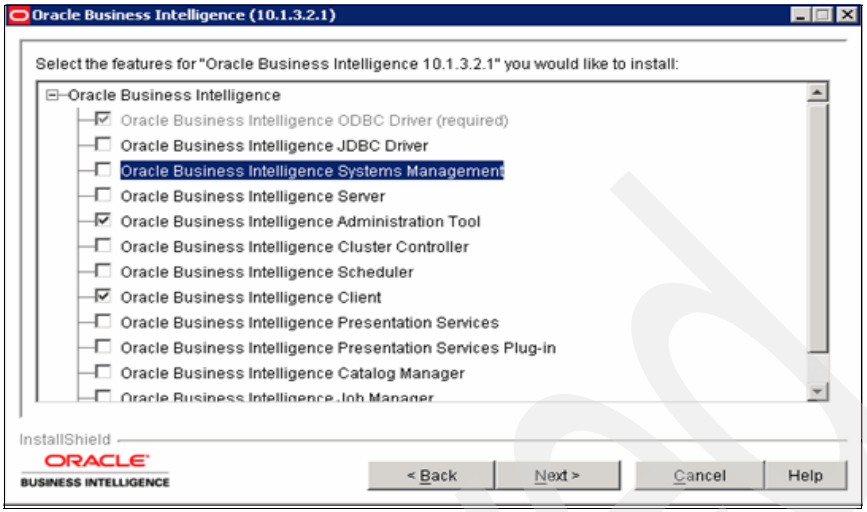


Figure 6-15 Custom choices

The option of which language for error messages in presented in Figure 6-16.

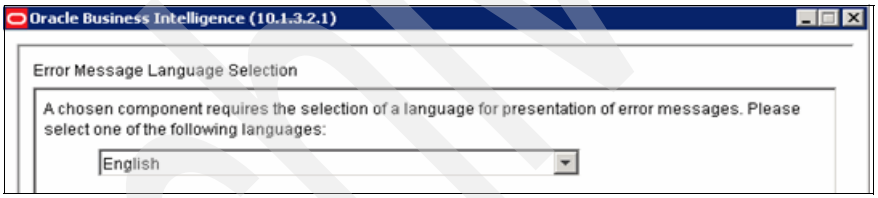


Figure 6-16 Language Choice

The summary of products displayed in Figure 6-17 confirms the choices we made.

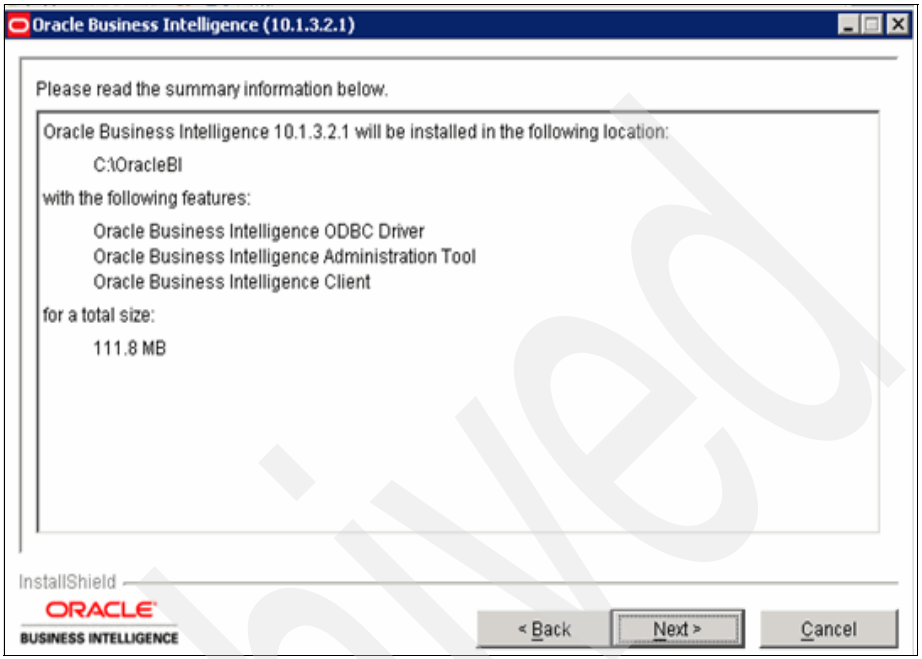


Figure 6-17 Summary of installation

After a short time, the installation is complete.

This project had a pre-built database repository to match the pre-built database that was the source of the import described in 6.2.1, “Populating the database” on page 100. The repository name needs to be defined on the Linux server supporting the Oracle BIEE Server component. The file is called NQSConfig.INI, and is located in $ORACLE_HOME/BIEE/server/Config>.

Replace the Rep_table with the name of your repository in the Repository section of the file:

```
###################################################################
[ REPOSITORY ]

Star = Rep_Table, DEFAULT;
###################################################################
```

When the definitions and updates are complete and the configuration is ready for production, this file should be updated as follows:

```
[ SERVER ]
READ_ONLY_MODE = YES
```

This will resolve some database performance issues.

Back on the Windows host, the ODBC data source also needs to be defined. This is accomplished by opening the Control Panel and selecting **Administrative tools** → **Data Sources (ODBC)**. The window shown in Figure 6-18 will appear.

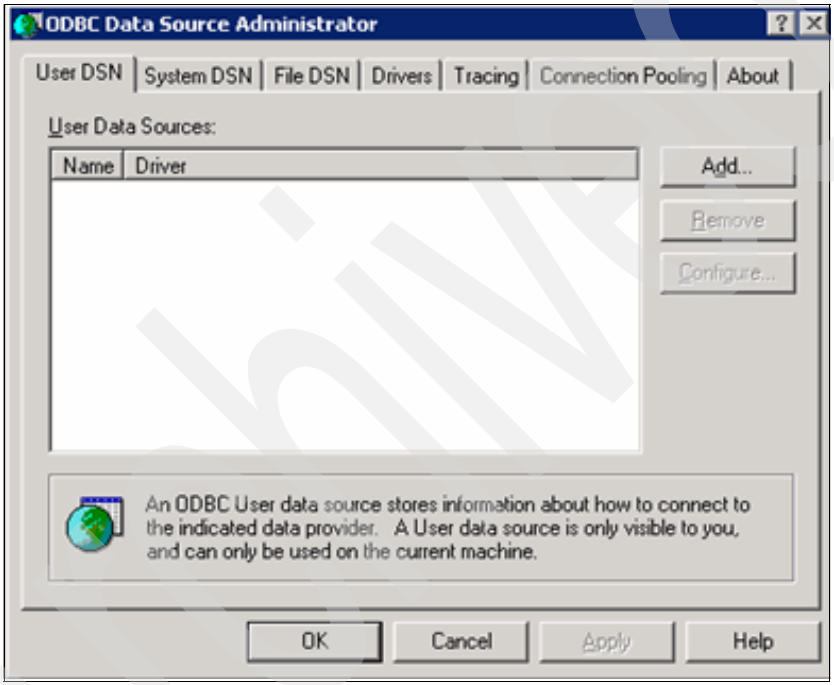


Figure 6-18 Data Source information

Select the **System DSN** tab. The window shown in Figure 6-19 is displayed.

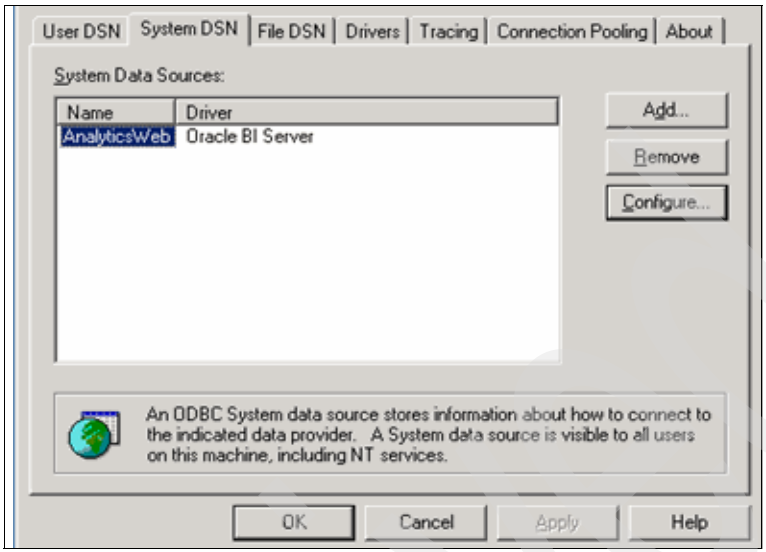


Figure 6-19 DNS System window

Highlight the **Analytics Web** and select **Configure**. Figure 6-20 displays. Here you update only the server field with the IP address or DNS resolvable host name of the Linux host with the Oracle BIEE Server. Select **Next** and save the changes.

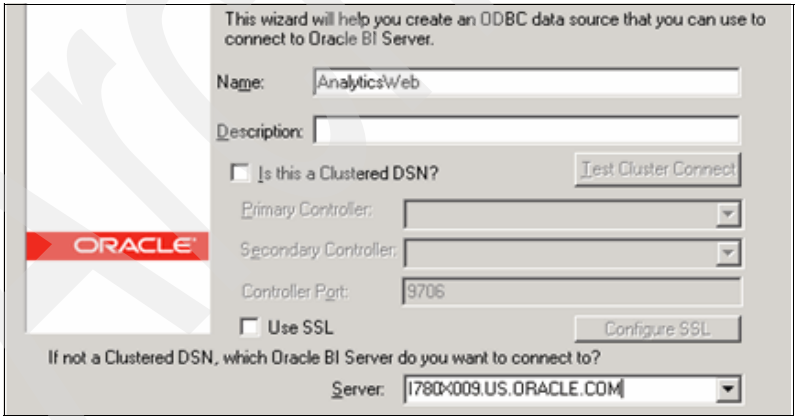


Figure 6-20 DSN configuration

The next step is to open the Oracle BIEE application on the Windows host from the Windows start menu. Once open, select the Open Remote Repository Blue folder. Figure 6-21 displays.

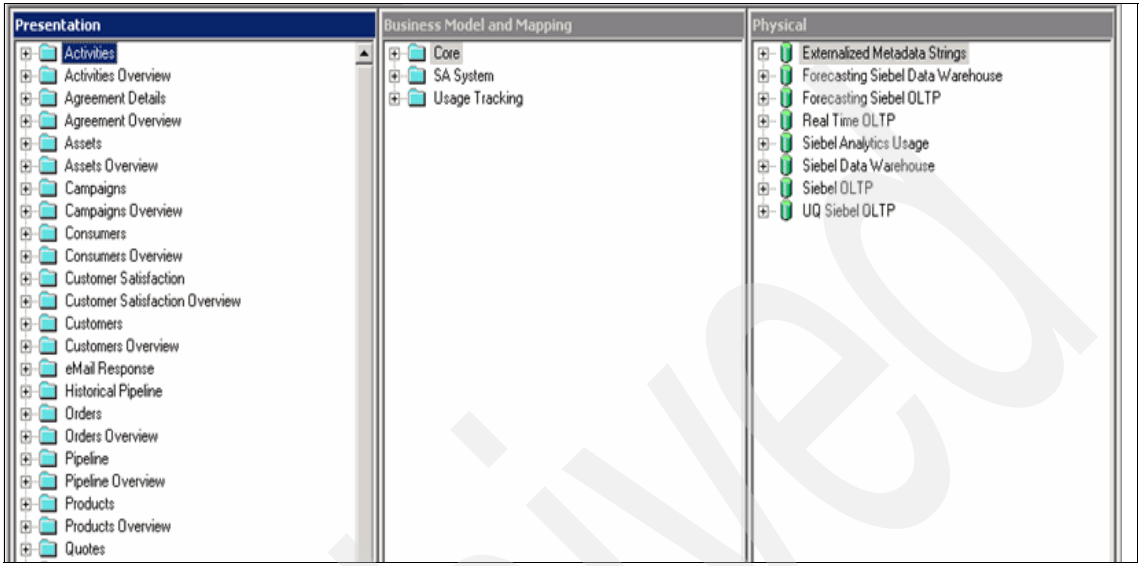


Figure 6-21 Oracle BI Administrator Tool for Data Warehouse

Highlight the Siebel Data Warehouse entry under the physical column. Right-click to update the properties selection. This will cause the Check Out Confirmation tab to be displayed, as shown in Figure 6-22. Select **Yes** to continue.

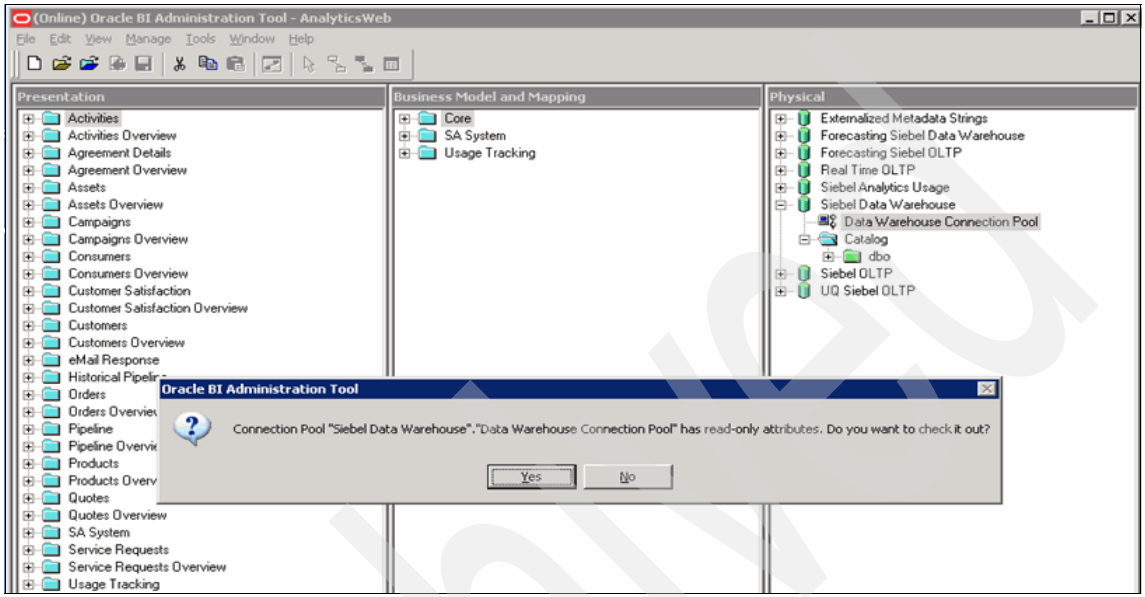


Figure 6-22 Oracle BI Administrator Tool - Analytic Web

Under the General tab, confirm that the Data source Database definition section is correct (see Figure 6-23).

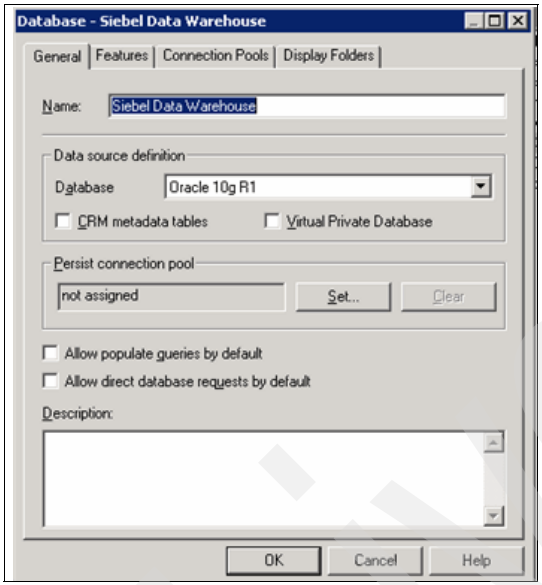


Figure 6-23 Siebel Data Warehouse

Under the Connection Pools Tab in Figure 6-23 on page 119, highlight the data warehouse connection pool selection below the Siebel data warehouse line. Right-click **Edit** to bring up the properties entry, shown in Figure 6-24. You will need to update the database login and password.

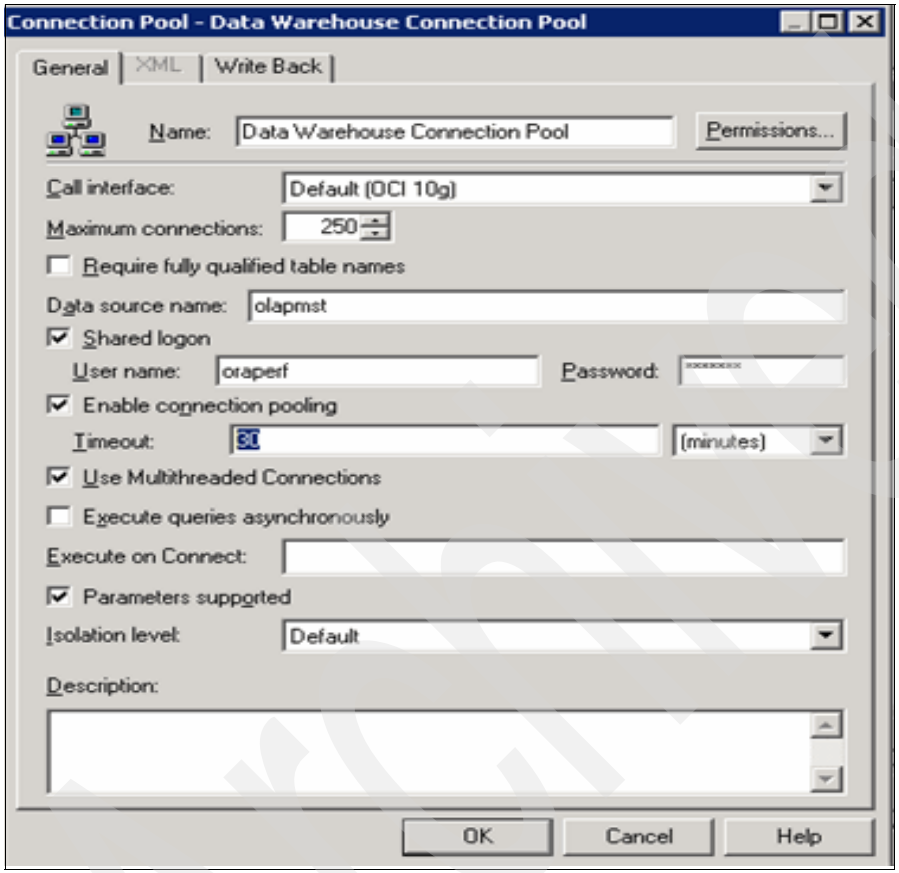


Figure 6-24 Data Warehouse Connection Pool

When you click **OK**, you will have to confirm your password.

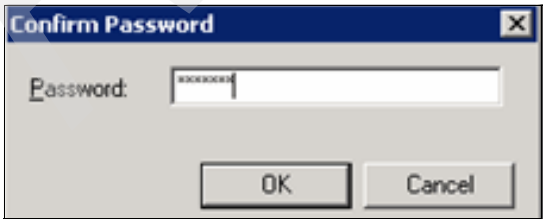


Figure 6-25 Password window

You will now have to check in the changes made by selecting the white paper icon with the red arrow, as shown in Figure 6-26. This will check in your changes and the red flags from the data warehouse, and the connection pools will have disappeared.

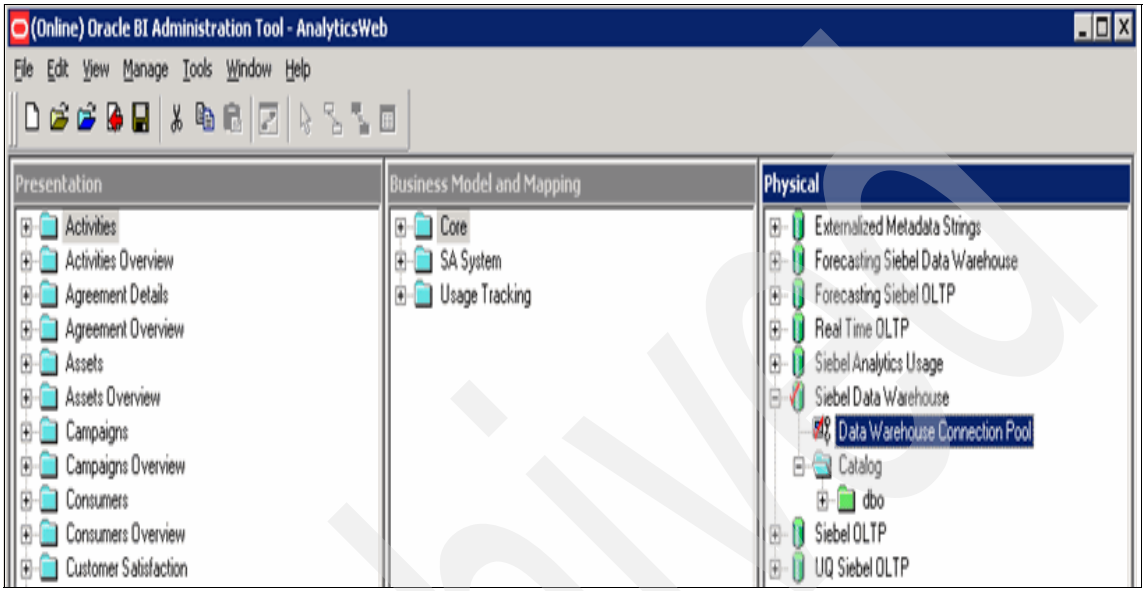


Figure 6-26 Check in window

At this point, the database is now connected to the application server. This can be validated by opening the database definitions and listing the tables and by executing the validation tests in the installation manual.

7

Setting up multiple Linux guests for Oracle Databases

This chapter describes the infrastructure that has been set up for running many Oracle Linux guests under VM. It is based on practical customer experiences. It covers:

- Using multiple VSwitches
- Using cloning to set up new images
- Using Flashcopy and z/OS to manage backups

7.1 Use of multiple VSwitches to support Oracle workloads

Network virtualization with z/VM supports the needs of large and complex network topologies. By using the VSwitch capabilities of z/VM, many physical networks can be extended into the virtual environment. LAN segments and VLANs configured on the physical switches can be extended into a VSwitch network, thereby eliminating the need for router virtual machines on z/VM, as shown in Figure 7-1 on page 125.

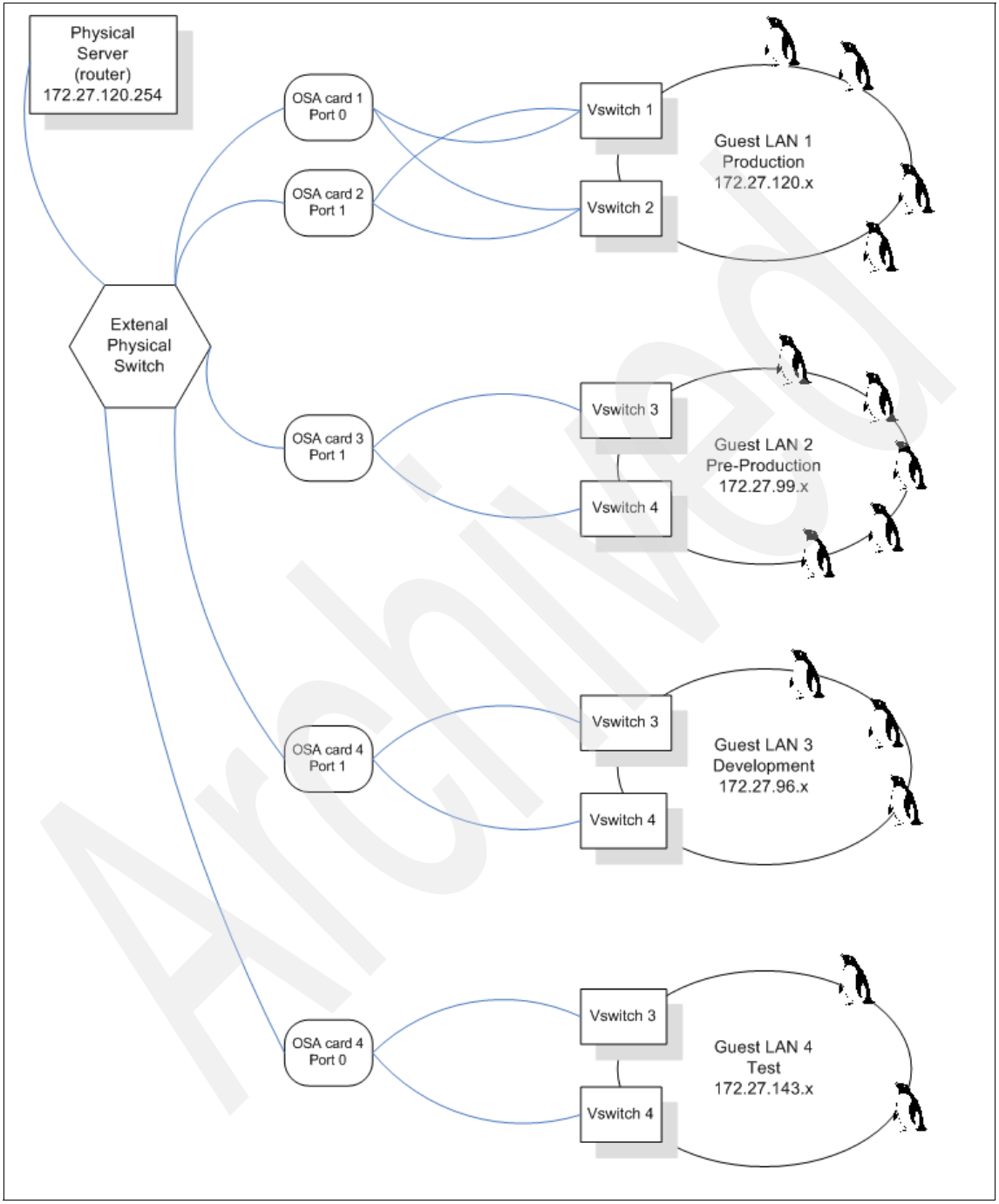


Figure 7-1 Sample of multiple VSwitches in a network

In Figure 7-1 on page 125, z/VM LPAR is shown with four different VSwitch networks serving the needs of production, pre-production, development, and test. None of these VSwitch networks connect, so they cannot route to each other within the LPAR. All of the VSwitch networks coexist without knowledge of each other.

All of the Linux machines on the VSwitches are running Oracle databases.

The VSwitch networks connect to the physical networks through OSA Express-2 cards as QDIO devices. Three device addresses are required for one set of QDIO devices: a read device, a write device, and a data transfer device.

VSwitches are defined in the SYSTEM CONFIG file read by CP at IPL time, or by the CP DEFINE VSWITCH commands. Virtual machines are allowed or denied participation in a VSwitch at the membership level by the SYSTEM CONFIG file, the CP SET VSWITCH GRANT or REVOKE commands, or can be protected by an ESM, such as RACF®.

Example 7-1 is an example of defining four VSwitches in a SYSTEM CONFIG file.

Example 7-1 VSwitch definitions

```
/************************************************************/
/*                VSwitch Definitions                       */
/************************************************************/

 Define VSwitch VSXXSAP2 IP Rdev D900 Controller * Vlan 999 Portt Trunk
 DEFine VSwitch VSXXVAP2 ip rdev D000 Controller * vlan 999 portt trunk
 DEFINE VSWITCH VSXXOAP2 IP RDEV DA00 DB00 CONTROLLER * VLAN 999,
 PORTT TRUNK
 DEFINE VSWITCH VSXXPAP2 IP RDEV E804 E904 CONTROLLER * VLAN 999,
 PORTT TRUNK
```

The VSwitches VSXXOAP2 and VSXXPAP2 have two sets of real devices defined, corresponding to their uses as production and pre-production networks. Device recovery is possible. The VSwitches VSXXSAP2 and VSXXVAP2 have only one set of rdevices. No device recovery is possible. These devices are attached to a controller machine when the virtual switch is created, or, given to an alternate controller machine in the unlikely event of a controller machine failure.

The controller machine does not participate in the movement of either layer two frames or layer three packets. The controller machine, although configured through the standard files used by the VM TCP/IP stack, is used for initialization and recovery purposes. Although confusing at first, it is very straightforward to work with the controller machines. In most cases, the IBM supplied controller

machines, DTCVSW1 and DTCVSW2, can handle all requirements for multiple VSwitch configurations.

Since no packet or frame movement takes place in a controller machine, it is a safe environment for controlling multiple networks, even networks that cannot see each other's data due to site security policies.

The directory entry for DTCVSW1 is shown in Example 7-2, and may be used exactly as provided in the z/VM installation: The user statements contain the defaults from installation time, and are satisfactory for production needs.

Example 7-2 DTCWSW1 entry

```
USER  DTCVSW1  DTCVSW1  32M  128M G
 INCLUDE TCPCMSU
 OPTION QUICKDSP SVMSTAT MAXCONN 1024 DIAG98
 IUCV *VSWITCH MSGLIMIT 65535
 LINK 5VMTCP30 491 491 RR
 LINK 5VMTCP30 492 492 RR
 LINK TCPMAINT 591 591 RR
 LINK TCPMAINT 592 592 RR
 LINK TCPMAINT 198 198 RR
*
MDISK 191 3390 22 005 VMXXAL MR RDTCVSW1 WDTCVSW1 MDTCVSW1
*DVHOPT LNK0 LOG1 RCM1 SMS0 NPW1 LNGAMENG PWC20070308 CRC7
```

The VSWITCH CONTROLLER ON statement from the DTCVSW1 TCP/IP configuration file is shown. The DTCVSW1 PROFILE may be used exactly as provided in the z/VM installation.

Example 7-3 VSWITCH CONTROLLER statement

```
;----------------------------------------------------
; Define the given virtual machine such that it can control CP-defined
; Virtual Switch connections to real LAN segments through OSA Express
; devices.
; ----------------------------------------------------

VSWITCH CONTROLLER ON
```

No DEVICE, LINK, HOME, or GATEWAY statements should be coded in a controller profile.

Figure 7-2 on page 128 shows the VSXXVAP2 VSwitch with the controller machine, several Linux virtual machines, and a VM TCPI/P machine. One Linux machine is shown with its virtual NICs and an IP address of 172.27.120.158.

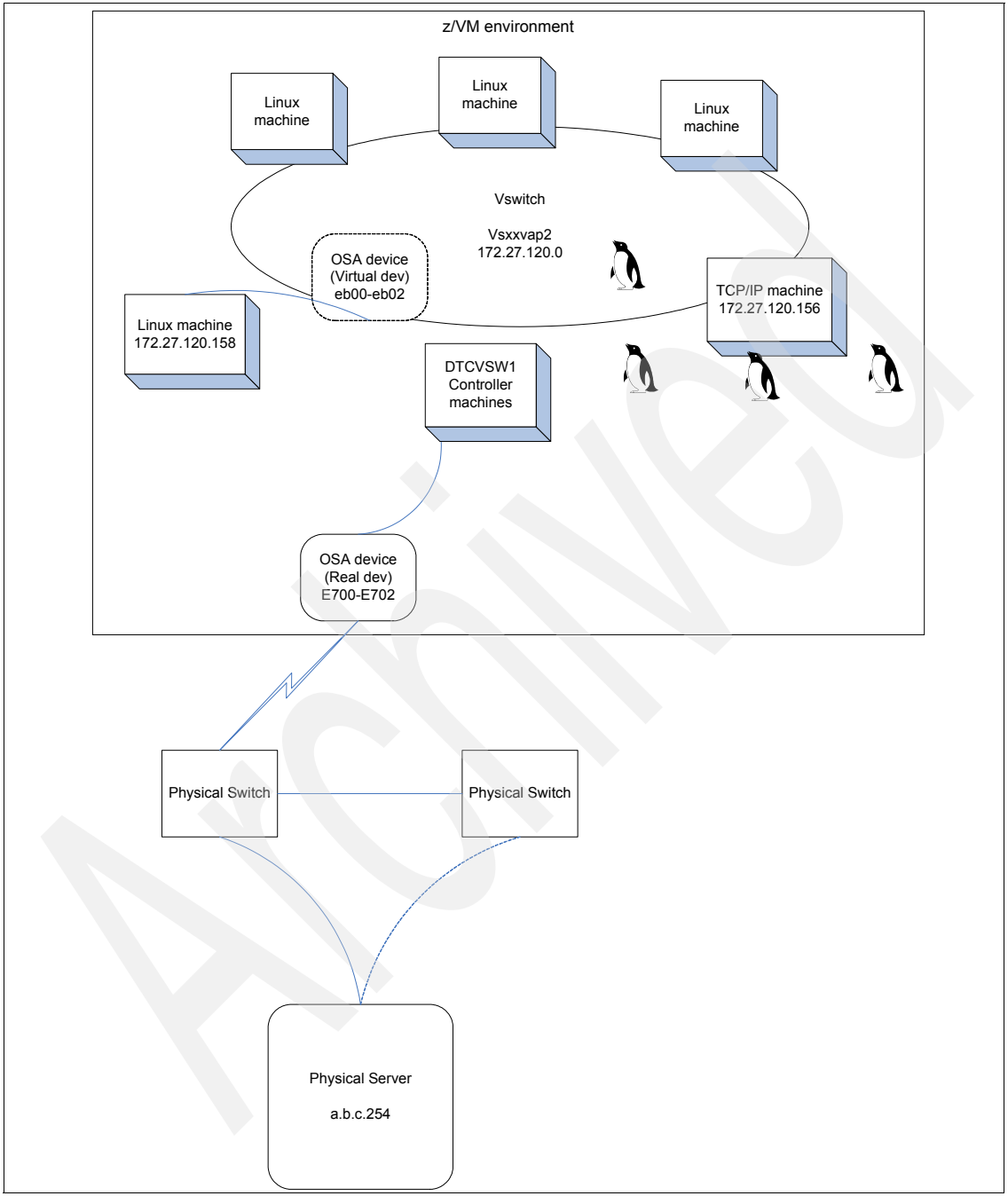


Figure 7-2 Sample network

A virtual machine connects to a VSwitch providing it has been given grant authority by a **cp** command or through an ESM with either a directory statement (NICDEF or SPECIAL) or **cp** commands (DEFINE NIC and COUPLE).

When using the directory NICDEF statement, the virtual NICs and coupling to the VSwitch are done by the one command. Example 7-4 shows an extract from a directory entry for the DBXXUA1 virtual machine.

Example 7-4 NICDEF statement

```
USER DBXXUA1 CHANGEIT 256M 1024M G
   INCLUDE SIGPLNX
 NICDEF EB00 TYPE QDIO DEVICES 3 LAN SYSTEM VSXXVAP2
```

The virtual machine is granted access to the VSwitch with the **set vswitch** command:

```
set vswitch vsxxvap2 grant dbxxua
```

The **nicdef** statement will define virtual NIC devices EB00, EB01, and EB02 as simulated QDIO, and couple them to the system defined VSwitch VSXXVAP2.

The remaining task is to configure the device in Linux as an Ethernet device. The device is configured in the /etc/sysconfig/network directory. The file name is ifcfg-qeth-bus-ccw- prepended to the fully qualified device name, in this case, 0.0.eb00, so: ifcfg-qeth-bus-cww-0.0.eb00.

A catenate (**cat**) command shows the contents of the file, as shown in Example 7-5.

Example 7-5 cat command

```
cat ifcfg-qeth-bus-ccw-0.0.eb00
BOOTPROTO='static'
UNIQUE=''
STARTMODE='onboot'
IPADDR='172.27.120.157'
MTU='1500'
NETMASK='255.255.255.0'
NETWORK='172.27.120.0'
BROADCAST='172.27.120.255'
REMOTE_IPADDR=''
```

Assuming that this file is from a master, and the clone has no other network access (we do not want to have a duplicate IP address), the 3270 interface will be needed to create the working configuration file. In the example, the IP address is changed from 172.27.120.157 to 172.27.120.158.

The Linux commands used are **cat**, **cp**, **mv**, **head**, **tail**, and **echo** (see Example 7-6). The **cp** command will make a copy of the configuration file to a new file called temptemp. The **head** command will take the first three lines of temptemp and by using output redirection will write it to a new file called newtemp. The **echo** command will write the line with the new IP address (172.27.120.157) to the newtemp file with the append redirection symbol. The escape character / is used so that the single quotes are sent to the output. The **tail** command will take the last five lines from temptemp and append it to newtemp using the append redirection symbol. The **cat** command will display the contents of the newtemp file with the updated IP address (172.27.120.158).

Example 7-6 cp command

```
cp ifcfg-qeth-bus-ccw-0.0.eb00 temptemp

cp ifcfg-qeth-bus-ccw-0.0.eb00 temptemp
#
head -n 3 temptemp > newtemp
head -n 3 temptemp > newtemp
#
echo IPADDR=\'172.27.120.158\' >> newtemp
echo IPADDR=\'172.27.120.158\' >> newtemp
#
tail -n 5 temptemp >> newtemp
tail -n 5 temptemp >> newtemp
#
cat newtemp
cat newtemp
BOOTPROTO='static'
UNIQUE=''
STARTMODE='onboot'
IPADDR='172.27.120.158'
MTU='1500'
NETMASK='255.255.255.0'
NETWORK='172.27.120.0'
BROADCAST='172.27.120.255'
REMOTE_IPADDR=''
```

The **mv** command will rename temptemp to the configuration file, ifcfg-qeth-bus-ccw-0.0.eb00. The **ifdown** command will take down the eth0 interface. The **ifup** command will bring p the eth0 interface reading from the new configuration file with the IP address of 172.27.120.158. See Example 7-7 on page 131 for more details.

Example 7-7 mv command

```
#
mv newtemp ifcfg-qeth-bus-ccw-0.0.eb00
mv newtemp ifcfg-qeth-bus-ccw-0.0.eb00
#
ifdown eth0
ifdown eth0
eth0
:     eth0      configuration: qeth-bus-ccw-0.0.eb00
#
ifup eth0
ifup eth0
eth0
eth0      configuration: qeth-bus-ccw-0.0.eb00
#
```

The **ifconfig eth0** command will show the interface with the new IP address. The ping command will show a ping to the VM TCP/IP stack address of 172.27.120.156. See Example 7-8 for more details.

Example 7-8 ifconfig command

```
ifconfig eth0
ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:00:00:00:00:07
  inetaddr:172.27.120.158 Bcast:172.27.120.255 Mask:255.255.255.0
inet6 addr: fe80::200:0:100:7/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:529727 errors:0 dropped:0 overruns:0 frame:0
           TX packets:245 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:79653878 (75.9 Mb)  TX bytes:181262 (177.0 Kb)
#
ping -c 1 172.27.120.156

ping -c 1 172.27.120.156
PING 172.27.120.156 (172.27.120.156) 56(84) bytes of data.
64 bytes from 172.27.120.156: icmp_seq=1 ttl=60 time=2.35 ms

--- 172.27.120.156 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.356/2.356/2.356/0.000
```

The CP QUERY VSWITCH command will show details about VSwitch vsxxvap2. The details are shown (including IP addresses) for the two connected machines, TCPIP and DBXXUA1 (Example 7-9).

Example 7-9 cp quert vswitch command

```
q vswitch vsxxvap2 details
VSWITCH SYSTEM VSXXVAP2 Type: VSWITCH Connected: 2    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    NONROUTER                 Accounting: OFF
  VLAN Unaware
  MAC address: 02-00-00-00-00-03
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  RDEV: E700 VDEV: E700 Controller: DTCVSW1
    VSWITCH Connection:
      MAC address: 00-09-6B-1A-77-16
      RX Packets: 13097      Discarded: 1          Errors: 0
      TX Packets: 18460      Discarded: 0          Errors: 0
      RX Bytes: 1194198              TX Bytes: 1406244
      Device: E702  Unit: 002   Role: DATA       Port: 0001  Index: 0001
  Adapter Connections:
       Adapter Owner: DBXXUA1 NIC: EB00  Name: dontcare
      RX Packets: 530553     Discarded: 0          Errors: 0
      TX Packets: 258        Discarded: 0          Errors: 0
      RX Bytes: 79788045             TX Bytes: 174856
      Device: EB02  Unit: 002   Role: DATA       Port: 0067  Index: 0003
      Options: Broadcast Multicast IPv6 IPv4
    Unicast IP Addresses:
      172.27.120.158       MAC: 02-00-00-00-00-07
    Multicast IP Addresses:
224.0.0.1MAC: 01-00-5E-00-00-01
Adapter Owner: TCPIP    NIC: FFFC  Name: UNASSIGNED
  RX Packets: 411587     Discarded: 0          Errors: 0
  TX Packets: 272964     Discarded: 0          Errors: 0
  RX Bytes: 57211398             TX Bytes: 58711047
  Device: FFFE  Unit: 002   Role: DATA       Port: 0077  Index: 0014
    Options: Broadcast Multicast IPv4 VLAN
    Unicast IP Addresses:
      172.27.120.156       MAC: 02-00-00-00-00-08
    Multicast IP Addresses:
      224.0.0.1            MAC: 01-00-5E-00-00-01
```

The **netstat arp** command will show the IP addresses that the OSA is aware of, including hosts on the VSwitch, and the physical host connected to the switch (172.27.120.254). The **netstat arp** commands are issued for the two virtual hosts (172.27.120.156 and 172.27.120.158) and for the one physical host (172.27.120.254). The controller virtual machine is aware of the arp table being maintained by the OSA device. See Example 7-10 on page 133 for more details.

Example 7-10 netstat command

```
netstat arp 172.27.120.156 tcp dtcvsw1
VM TCP/IP Netstat Level 530

ARP Age: 5

Querying ARP cache for address 172.27.120.156

Adapter-maintained data as of: 12/18/07 21:53:27

Link VSEESGS1D900LINK: QDIOETHERNET: 00096B1A7AE3 IP: 172.27.120.156
Ready; T=0.01/0.01 21:55:42
netstat arp 172.27.120.158 tcp dtcvsw1
VM TCP/IP Netstat Level 530

ARP Age: 5

Querying ARP cache for address 172.27.120.158

Adapter-maintained data as of: 12/18/07 21:55:42

Link VS120SGSE700LINK: QDIOETHERNET: 00096B1A7716 IP: 172.27.120.158
Ready; T=0.01/0.01 21:55:59
netstat arp 172.27.120.254 tcp dtcvsw1
VM TCP/IP Netstat Level 530

ARP Age: 5

Querying ARP cache for address 172.27.120.254


Adapter-maintained data as of: 12/18/07 21:55:59

Link VSEESGS1D900LINK: QDIOETHERNET: 080020E46479 IP: 172.27.120.254
Link VS120SGSE700LINK: QDIOETHERNET: 080020E46479 IP: 172.27.120.254
Ready; T=0.01/0.01 21:56:10
```

z/VM provides a great deal of flexibility in the configuration of virtual networks. In Figure 7-1 on page 125, the production and pre-production networks have two sets of OSA QDIO devices: one active and one standby. The test and development networks only have one set of OSA qdio devices, and will not recover from a cable pull error, or other (unlikely) OSA failures.

In z/VM V5.3.0, link aggregation of up to eight OSA devices for use by one VSwitch is possible, and will provide facilities for load balancing and high bandwidth by grouping OSA devices together.

Large scale network virtualization is readily implemented in a z/VM and Linux environment.

7.2 Setting up a cloning architecture

One of the major strengths of z/VM as a virtualization host for multiple Linux servers is a rich set of tools for Linux server replications.

There are several methods of cloning in z/VM, ranging from manually creating directory entries, copying disks, and updating Linux configuration files by hand, in-house automated solutions, to vendor products.

The solution presented here consists of multiple steps from installation to usable master. The coding was done using CMS REXX™ and PIPELINEs and has an automated front end to DIRMAINT on multiple systems and RACF.

A large client runs Oracle databases on nearly 200 Linux servers. In this environment, replication of several master images is an important part of systems administration. A cloning methodology was developed based on using SUSE SLES. The methodology uses very few masters. The masters contain a purified Linux distribution and a pre-installed Oracle.

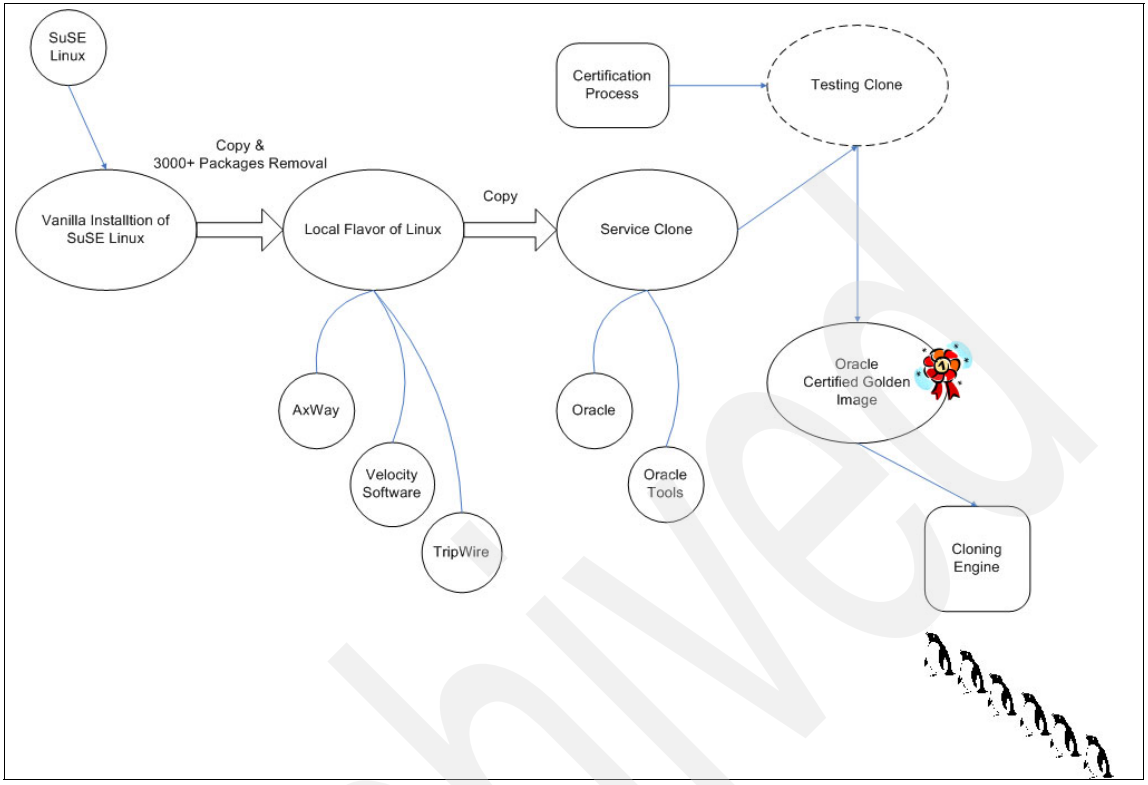


Figure 7-3 Replication architecture

The major steps, as shown in Figure 7-3, are:

1. Installation is performed on a SUSE SLES 9 or 10 package from a standard install from a directory tree that resides in a z/VM Linux virtual machine. Service packs are installed as well.

   After the install is completed, many packages that have been deemed to be unnecessary are removed. Package removal is a labor intensive step with around 3,000 packages being removed.

   This initial Linux system, minus the unnecessary packages, is then copied over to a new set of disks in preparation for the next step of installing add on packages, products, and local tailoring.

2. Installation of add-on packages and products and local tailoring.

   The additional packages provide software for performance monitoring from Velocity Software, security compliance and change notification with the Tripwire package, and the approved method of file transfer using Axway.

Local tailoring involves the installation of scripts used in the cloning process, changes to the FSTAB to conform to the disk standards in use, and configuration skeleton files for the network interfaces.

The tailored system is then copied to a new set of disks in preparation for the next step, the Oracle installation.

3. Oracle installation.

   Oracle is installed. The file systems where Oracle is installed are shown in Figure 7-4.

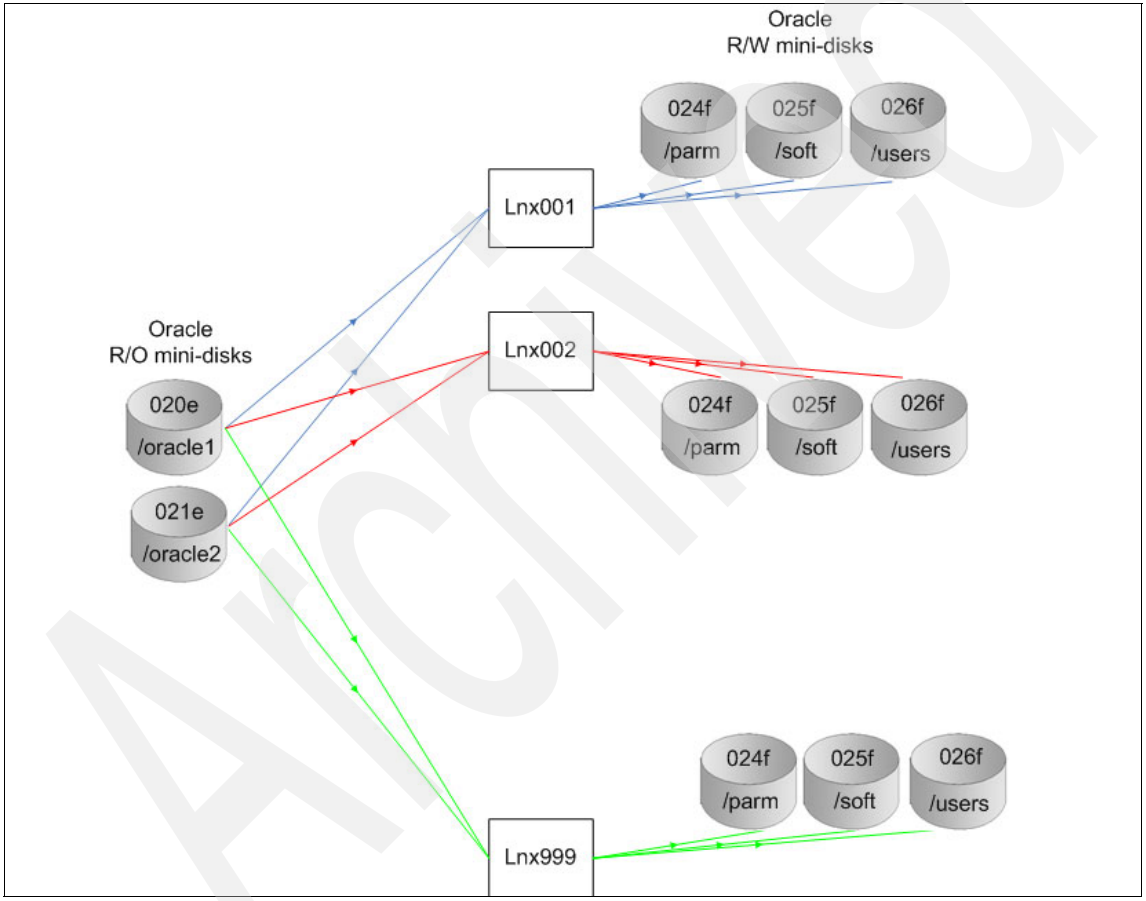


Figure 7-4 Oracle shared read-only and non-shared read-write disks

4. Extensive testing by different departments

   The Oracle system is then copied over to a new set of disks in preparation of the testing system. The testing system is turned over from the builders to Linux sysadmins, z/VM systems programmers, Oracle DBAs, the automation group, and the security administrators.

   Corrections and changes are reviewed, approved, and then tested. When all tests have been passed, this certified system is now used as a master for input to the cloner.

5. The approved golden image is made ready as the master input for cloning. The approved and certified golden image is used as input for the cloner.

   The replication code engine is done in a CMS server in the service zone LPAR. The service zone LPAR is configured to recognize all DASD devices on the z/9 CEC. The coding engine is written in REXX, and interfaces with CP, the VM directory through DIRMAINT commands, and RACF commands.

   DIRMAINT is configured to send commands to remote systems. Example 7-11 shows a command issued from the SERVICE LPAR to the APPLIC LPAR.

Example 7-11 Dirmaint commands

```
query userid
MAINT    AT SERVICE
Ready; T=0.01/0.01 14:43:02
dirm to applic for sigdk01 get nolock
DVHXMT1191I Your GET request has been sent for processing.
Ready; T=0.02/0.02 14:43:07
 From APPLIC(DIRMAINT): DVHREQ2288I Your GET request for SIGDK01 at * has
 From APPLIC(DIRMAINT): DVHREQ2288I been accepted.
 From APPLIC(DIRMAINT): DVHGET3305I Entry SIGDK01 sent, no lock attempt was
 From APPLIC(DIRMAINT): DVHGET3305I made.
RDR FILE 0024 SENT FROM RSCS     PUN WAS 1339 RECS 0011 CPY  001 A NOHOLD NOKEEP
 DMTAXM104I File (2471) spooled to MAINT -- origin APPLIC(DIRMAINT) 01/08/08
 14:43:07 EST
 From APPLIC(DIRMAINT): DVHREQ2289I Your GET request for SIGDK01 at * has
 From APPLIC(DIRMAINT): DVHREQ2289I completed; with RC = 0.
```

The RSCS service machine on both LPARs is used to transport the messages. Figure 7-5 shows the inter-LPAR communication used by DIRMAINT.

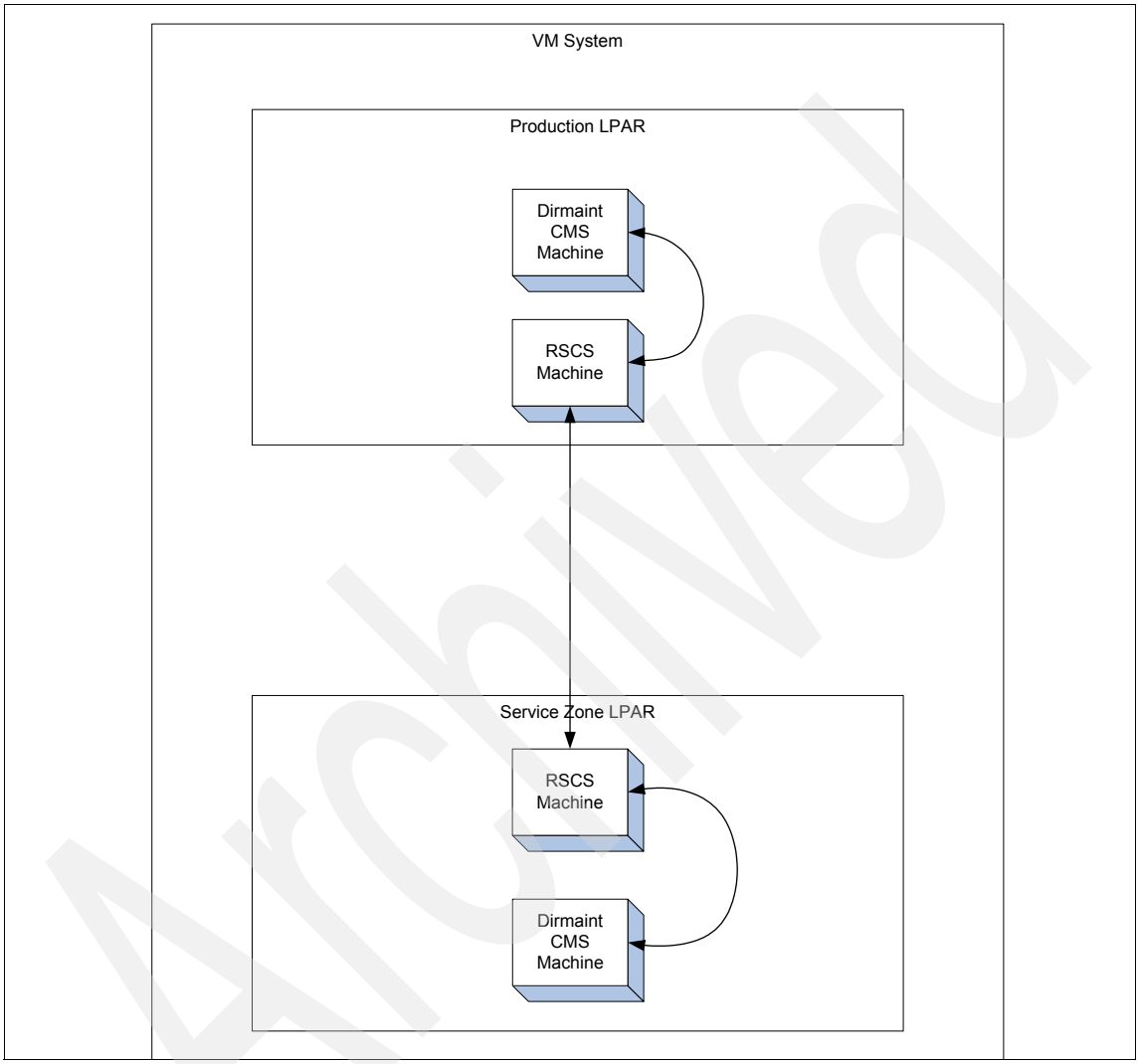


Figure 7-5 RSCS to RSCS inter LPAR communication on a z/VM system

The RACF database is shared between the production LPARs and the service zone LPAR, as shown in Figure 7-6 on page 139.

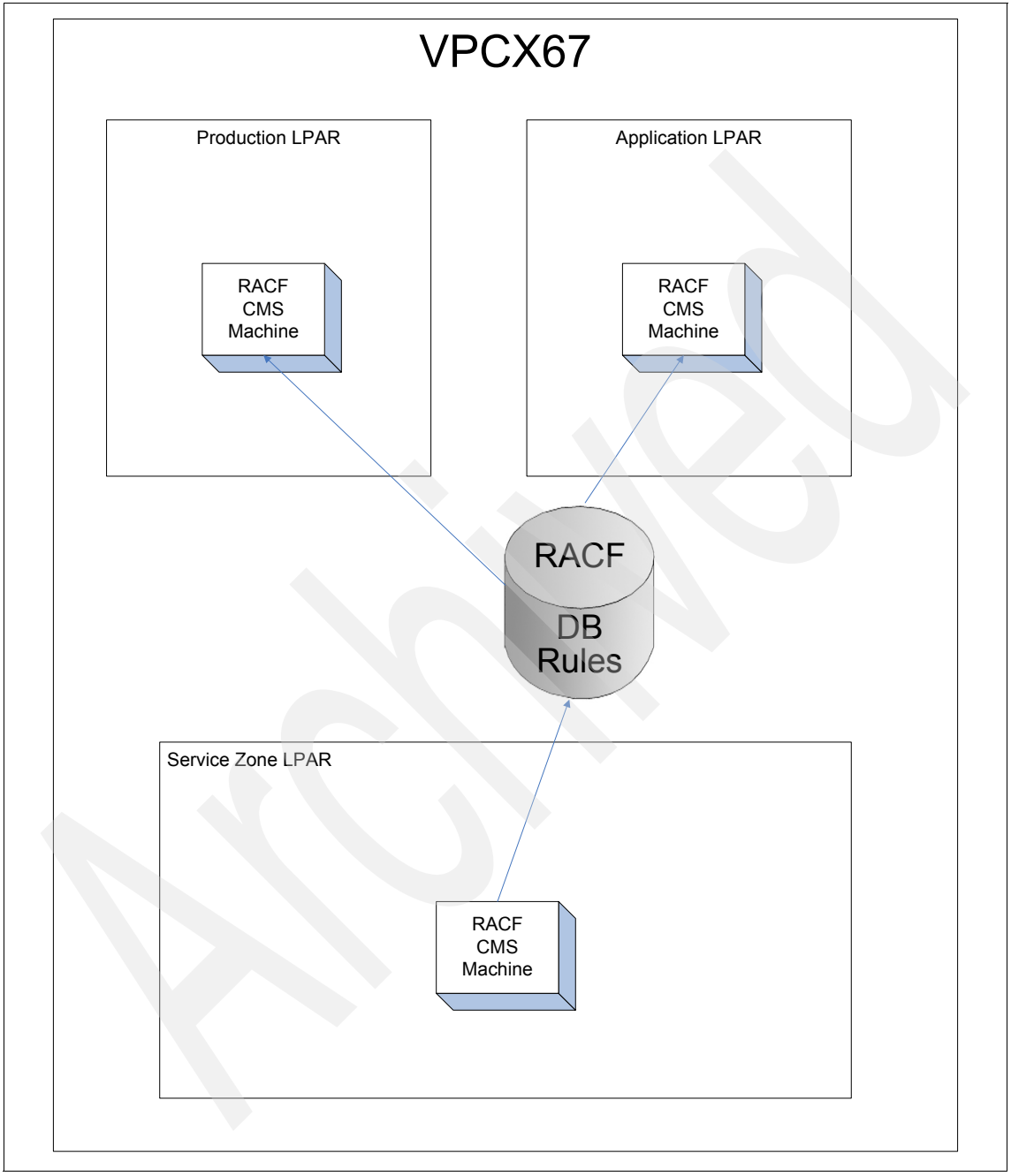


Figure 7-6 RACF shared databases

The replication engine is based on sharing large portions of the Linux hierarchical file system, and modifying the root file system structure to include smaller disks. The small disks are used for local parameters for a Linux host, containing customized configuration files. These smaller file systems are then bind mounted into the standard Linux file systems structure.

Figure 7-7 shows the disk architecture used by a replicated server.

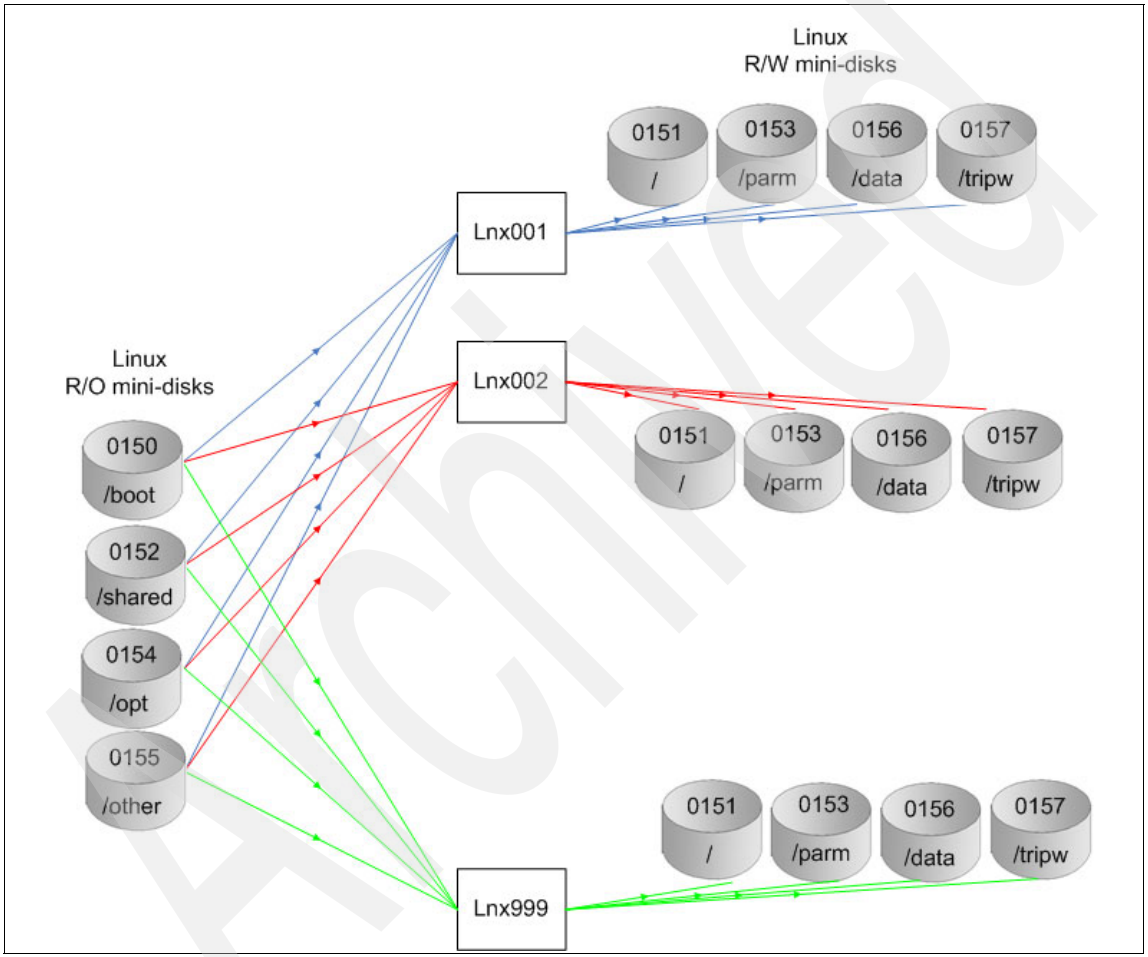


Figure 7-7 DASD configuration for a Linux Server

The R/O disks are linked through CP directory LINK statements, and are mounted r/o by Linux at startup time. The tailoring of the file system is done in the master image during the image preparation.

Linux bind mounts allow mounted parts of the file system to be mounted at another location. Bind mounts can then be used in combination with VM minidisks to graft the r/w smaller parm disks into the appropriate location in the file system. This technique combines disks with different attributes (such as a r/w disk and a r/o disk) into a seamless view of the file system. Bind mounts can be used to place a tailored file from a parm disk into the standard location of the hierarchical file system.

For example, the 152 minidisk is linked R/O by the replicated Linux server. The 152 is mounted as /SHARED, and then various parts /SHARED are bind mounted into the /usr hierarchy. The /usr hierarchy is, of course, an excellent candidate for sharing, as its contents are shared and static. Example 7-12 shows how 152 (dasdc) is bind mounted.

Example 7-12 Mount commands for disk 152

```
send cp lsigapa4 query virtual 152
LSIGAPA4: DASD 0152 3390 VPDX03 R/O       2502 CYL ON DASD  3664 SUBCHANNEL = 000B

sendit lsdasd | grep dasdc
LSIGAPA4: lsdasd | grep dasdc
LSIGAPA4: 0.0.0152(ECKD) at ( 94:  8) is dasdc      : active at blocksize 4096,
450360 blocks, 1759 MB
LSIGAPA4: LSIGAPA4:/ #

sendit mount | grep dasdc

LSIGAPA4: mount | grep dasdc
LSIGAPA4: /dev/dasdc1 on /SHARED type ext2 (ro)
LSIGAPA4: LSIGAPA4:/ #

sendit mount | grep SHARED

LSIGAPA4: mount | grep SHARED
LSIGAPA4: /dev/dasdc1 on /SHARED type ext2 (ro)
LSIGAPA4: /SHARED/lib on /lib type none (ro,bind)
LSIGAPA4: /SHARED/lib64 on /lib64 type none (ro,bind)
LSIGAPA4: /SHARED/usr on /usr type none (ro,bind)
LSIGAPA4: /SHARED/bin on /bin type none (ro,bind)
LSIGAPA4: /SHARED/sbin on /sbin type none (ro,bind)
LSIGAPA4: /SHARED/logiciels on /logiciels type none (rw,bind)
LSIGAPA4:
```

The SENDIT EXEC is a short REXX routine that sends mixed case commands to a Linux host, LSIGAPA4.

In virtualization strategies employing techniques that exploit features of both the virtualized system (Linux) and the host system (z/VM) combine the best of both worlds. Using facilities inherent in CP, CMS, and Linux facilitates excellent methods of replication.

7.3 Using FlashCopy so backups can be managed by z/OS

This section describes one approach for managing backups of Oracle databases.

The methods shown here are for EXT2 or EXT3 file systems. If you are using the Oracle ASM file system, other considerations must be taken into account. ASM issues are not covered in this section.

7.3.1 Backups

There are many approaches for backing up and restoring Linux virtual machines on the IBM System z server. This includes Linux file level backups through open source packages, VM backups through system utilities or vendor products, and backing up from z/OS systems.

While lacking some capabilities, such as the ability to perform Linux file level archiving, performing backups and restores with the aid of a z/OS system is a popular choice among z/VM and Linux installations. Many z/VM shops have limited (or no) access to tape drives, tape vaults, or robotic tape systems.

Many z/VM and Linux installations have existing mainframes running z/OS, complete with backup/restore software, procedures, and policies.

There are two configurations of z/OS reaching out to the DASD owned by a z/VM system:

1. A z/OS LPAR is in the same CEC as the z/VM LPAR(s). The z/OS LPAR would have access to the z/VM DASD through the coding of the devices in the IOCDS.
2. The z/VM LPAR(s) and the z/OS LPAR are on separate CECs. In this case, the DASD storage unit itself has to be dual cabled into both CECs. Each CEC has an IOCDS coded that is configured to see the same DASD.

A large client with Oracle virtual machines utilizes a configuration of two, separate CECs. Figure 7-8 on page 143 shows the architecture of the backup solution.

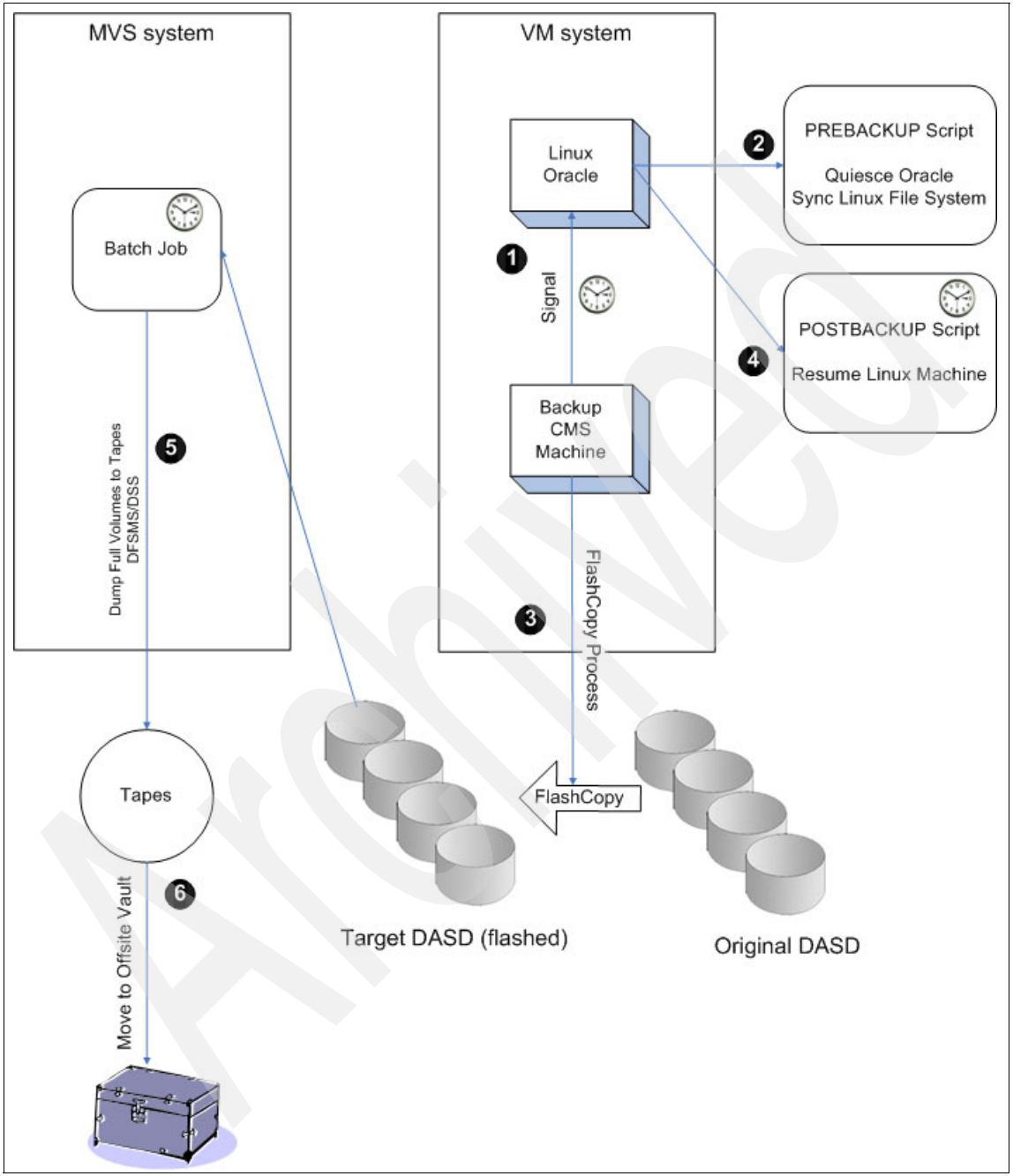


Figure 7-8 Backup architecture

The steps of the backup are:

1. A CMS service machine in the z/VM LPAR issues console commands as the root user to each Linux machine at a preset time from a remote system. The remote system is a service LPAR.
   a. The service machine, SIGBKUP, is started by a timer based program such as WAKEUP.
   b. Using the CP secondary console input facility, issue commands through the CP SEND command to each Linux guest that is to be backed up.
2. The Linux machine processes the console command PREFLASH. PREFLASH is a bash script that:
   a. Quiesces Oracle. Once a week, the database is stopped with **dbora stop**. Other times, a hot backup is performed and the database is stopped with the command **alter database begin backup**.

      Note: When quiescing Oracle with **alter database begin backup** and the database spans multiple ECKD volumes using the ext3 file system, there is a small window for I/O inconsistency. There is a slight chance that the database will not be restored to a good state. However, using **alter database suspend** instead of **alter database begin backup** will provide a clean environment in this situation.

   b. Issues the **sync** command several times. The **sync** command is issued several times in each machine to ensure that the disks are updated. Linux will cache file system updates in memory, and the **sync** command will write any data buffered in memory out to disk
3. The CMS server machine on the service LPAR will now:
   a. Link to the full volumes of all DASDs containing the Linux machine minidisks. Minidisks containing Oracle data for production machines are backed up daily. Minidisks containing system data such as executables are backed up less frequently (lower change rate).

      The backups are done by pools of machines, that is, production, development, lab, and so on.
   b. Link to the targeted pair of the full volumes.
   c. Perform a FLASHCOPY from the source volume to the target volume starting from cylinder 1 to the last cylinder. Cylinder 0 is not required for the FLASHCOPY, as no minidisks start on cylinder 0, and this way the volume label on the target is preserved.

A lookup is done for each minidisk belonging to the Linux machine, and a list of full volumes is prepared. SIGBKUP links to the SYSDUMP1 minidisk at virtual address 380E for the source, and 388E for the target. Example 7-13 shows the list.

Example 7-13 List of minidisks

```
VPCX81 380E 3390 SYSDUMP1 380E 0     3338  3339  FC380E ZDONXXX1 CPCMS
VPCX81 380E ---- PSSSDPB8 0156 1     350   350   FC380E ZDONXXX1 LINUX
VPCX81 380E ---- PSISSPB8 024F 351   510   160   FC380E ZDONXXX1 LINUX
VPCX81 380E ---- PSSSDPB8 025F 511   1066  556   FC380E ZDONXXX1 LINUX
VPCX81 380E ---- PSSSDPB8 026F 1067  1516  450   FC380E ZDONXXX1 LINUX
VPCX81 380E ---- PMRCDCA1 0150 1517  2104  588   FC380E ZDONXXX1 LINUX
VPCX81 380E ---- PMRCDCA1 0153 2105  3281  1177  FC380E ZDONXXX1 LINUX
```

Each full volume has a “twin” volume that is the target of the FLASHCOPY. Figure 7-9 on page 146 shows the source and target disks linked from the SIGBKUP CMS server machine, and the FLASHCOPY command.

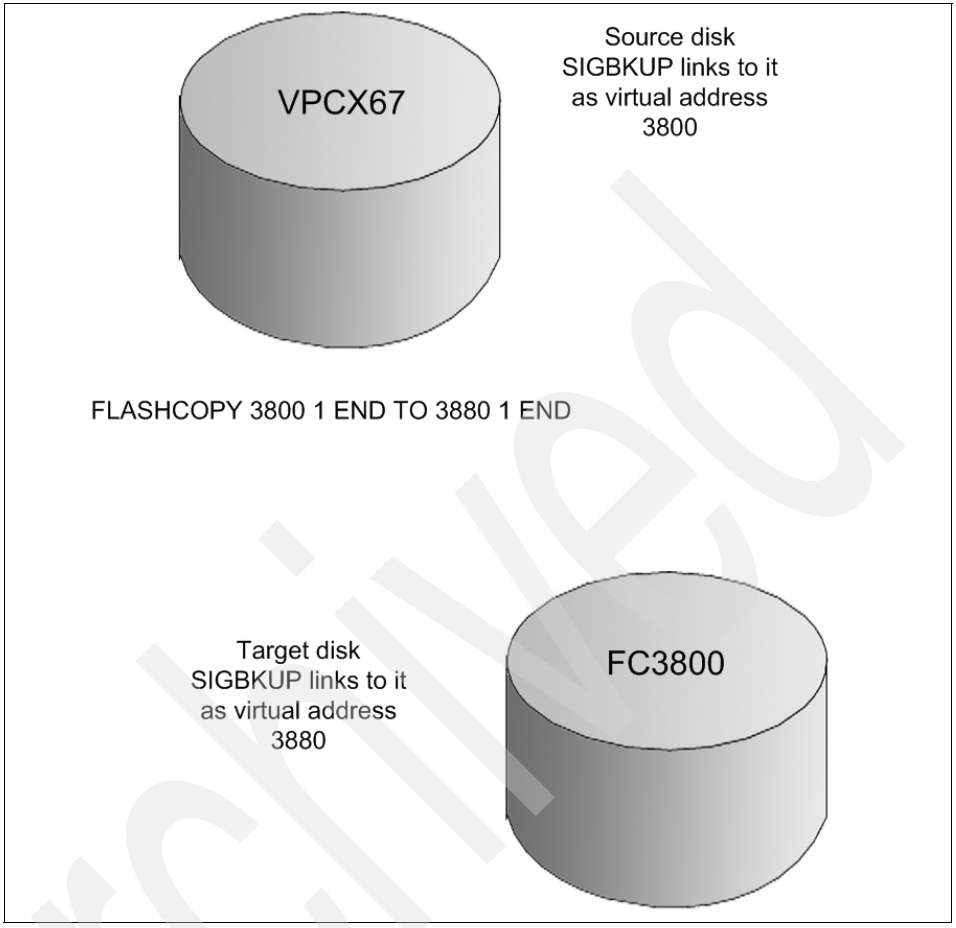


Figure 7-9 FLASHCOPY for source to target volume

4. When the FlashCopy® is complete, the CMS service machine in the z/VM LPAR issues a console command as the root user to resume Oracle. This is done using the POSTBACKUP script. The Linux machines processes the POSTBACKUP script to resume Oracle. The console commands start up the database after it is stopped, or **`alter database end backup`** is run in the case of a hot backup.
5. A batch job is started on the z/OS LPAR to back up the twinned volume to tape. The list of volumes to back up is shipped from the CMS server machine to z/OS.
6. z/OS using DFSMS/DSS backs up the flash target to tape. Tapes are stored offsite.

The entire process of quiescing the database, FlashCopying the disks, and resuming the database typically takes no more than one minute per Linux machine. The Oracle machines are backed up in groups, such as production, test, development, and so on. Production machines are backed up on a daily basis.

The backup to tape from the z/OS can take several hours, but, on the z/VM side all the Oracle database virtual machines are up and available. The z/OS backup is done on a full volume basis. The full volumes contain minidisks, some volumes have multiple minidisks, and other volumes contain one large minidisk. When planning z/VM storage, a volume should contain minidisks for the same virtual machine as much as possible. This will provide a general scheme of locality of reference for the FlashCopy and tape backup process by having the minidisks of a Linux server on as few real volumes as possible. A real volume should be defined with minidisks from the same Oracle Linux server.

7.3.2 Restore

Using the backup method already described, the restore method will restore all volumes backed up to a standby Linux virtual machine running Oracle. The standby Linux machine does not provide any real time Oracle services. Once full volumes are restored, and the file systems are available on the standby Linux machine running Oracle, Oracle is started with the backed up databases. Files are FTPed to the original Linux machine. Files can contain entire databases, portions of databases, or records. Figure 7-10 on page 148 shows the architecture of the restore method.

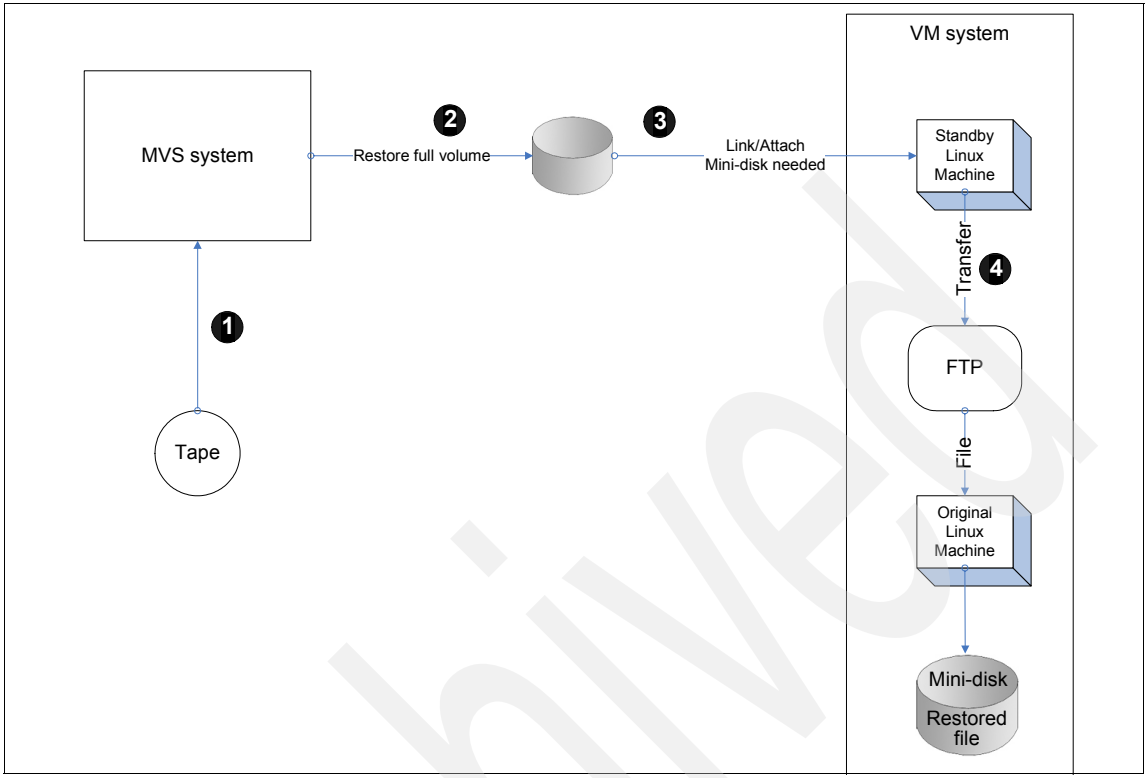


Figure 7-10 Restore architecture

As shown in Figure 7-10, the steps are:

1. A tape is recalled on z/OS.
2. The necessary full volume(s) are restored using DFSMS/DSS.
3. Minidisks are assigned to the standby Linux virtual machine.
4. Files are transferred using FTP to the original machine requiring the restore.

Although this method seems cumbersome and lacks sophistication, it does work quite well. Production environments have used this method for restoring old databases ranging from the record level.

Part 2

Using Oracle on Linux for System z

This part of the IBM Redbooks publication describes our experiences using special features of Oracle and IBM products as examples of things that could be used in a production environment.

We describe how we set up these components as follows:

- Oracle Data Guard with a primary and standby database on Linux for System z
- IBM WebSphere Application Server to connect to an Oracle Database on Linux for System z
- Oracle Data Pump to move an Oracle database from z/OS to Linux for System z
- VM LDAP and RACF to authenticate users for an Oracle database on Linux for System z

8

Using WebSphere Application Server with Oracle 10gR2

Chapter 11, “Using WebSphere with Oracle9i”, in *Experiences with Oracle for Linux on zSeries*, SG24-6552, discussed WebSphere Application Server V4.0 using an Oracle9i database as the repository for WebSphere Application Server. The WebSphere Application Server repository contains copies of the individual component configuration documents, but from WebSphere Application Server Version 5 onwards, the configuration information has been stored in XML files instead of in a relational database. The application server's Admin service is used to manage the configuration data in the XML files and to make sure it is consistent at runtime.

To demonstrate the use of WebSphere Application Server with an Oracle database in this book, the IBM Trade 6 Performance Benchmark Application was installed with WebSphere Application Server V6.1 and the application was configured to use an Oracle 10.2.0.2 database. Trade 6 is a WebSphere Application Server sample application that is publicly available from IBM. The Trade 6 benchmark models an electronic stock brokerage application providing Web-based online securities trading. It is implemented as a 3-tier J2EE Web application, utilizing an application server, a database server, and a Web server.

8.1 WebSphere Application Server V6.1, Oracle 10.2.0.2 DB, and the Trade 6 Performance sample application

An existing Oracle database was used to install the Trade 6 application's data, so this chapter does not cover the database installation process that is described in *Experiences with Oracle?® 10gR2 Solutions on Linux for IBM System z*, SG24-7191. The following steps are covered in this chapter:

- Install IBM WebSphere Application Server V6.1, referred to as base.
- Install IBM HTTP Server V6.1.
- Install, configure, and execute IBM Trade 6 Application.

8.2 Install WebSphere Application Server V6.1(base) and IBM HTTP V6.1

Both the WebSphere Application Server and the IBM HTTP Web server were installed on the same Linux system, known as *linux25*. The Oracle 10.2.0.2 database also was installed on linux25. This configuration is sufficient for the feasibility demonstration here, but not recommended for production.

The launchpad console is the starting point for installing the WebSphere Application Server and the HTTP Web server. The launchpad is a Web application, so a supported Web browser is needed to run it. The following is the browser that was used for the installation on Linux for System z. Also, the installation was done as the root user.

```
   [root@linux25 ~]# /usr/bin/firefox -version
Mozilla Firefox 1.5.0.3, Copyright (c) 1998 - 2006 mozilla.org
[root@linux25 ~]# export BROWSER=/usr/bin/firefox
[root@linux25 ~]# ./launchpad.sh
```

The launchpad program is in the root directory of the product disc or the downloaded image. The launchpad is named launchpad.sh on Linux for System z. Executing launchpad.sh results in the window shown in Figure 8-1.

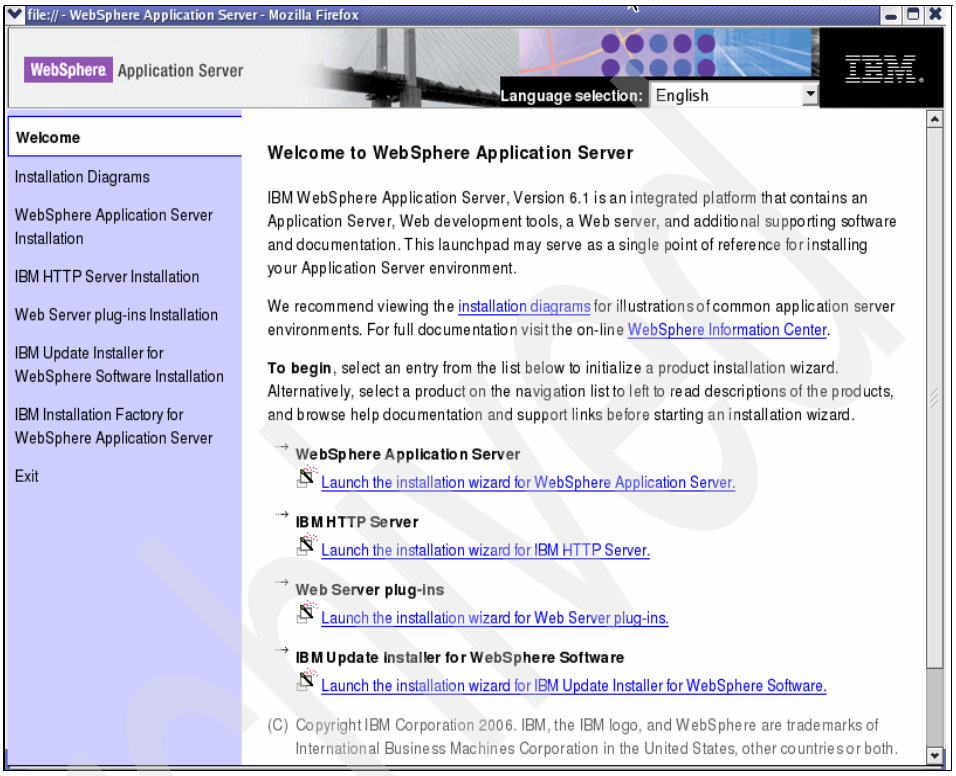


Figure 8-1 Welcome window for WebSphere Application Server

The actual installation is straightforward for installing the WebSphere Application Server and the IBM HTTP Server, but we should note two things. First, a prerequisite check failed with a warning message as the installer was checking for Red Hat Enterprise Linux 4 Update 2 and linux25 is at Update 5. Everything worked fine at the Update 5 level. Second, when the window shown in Figure 8-2 appeared during installation, we enabled administrative security.

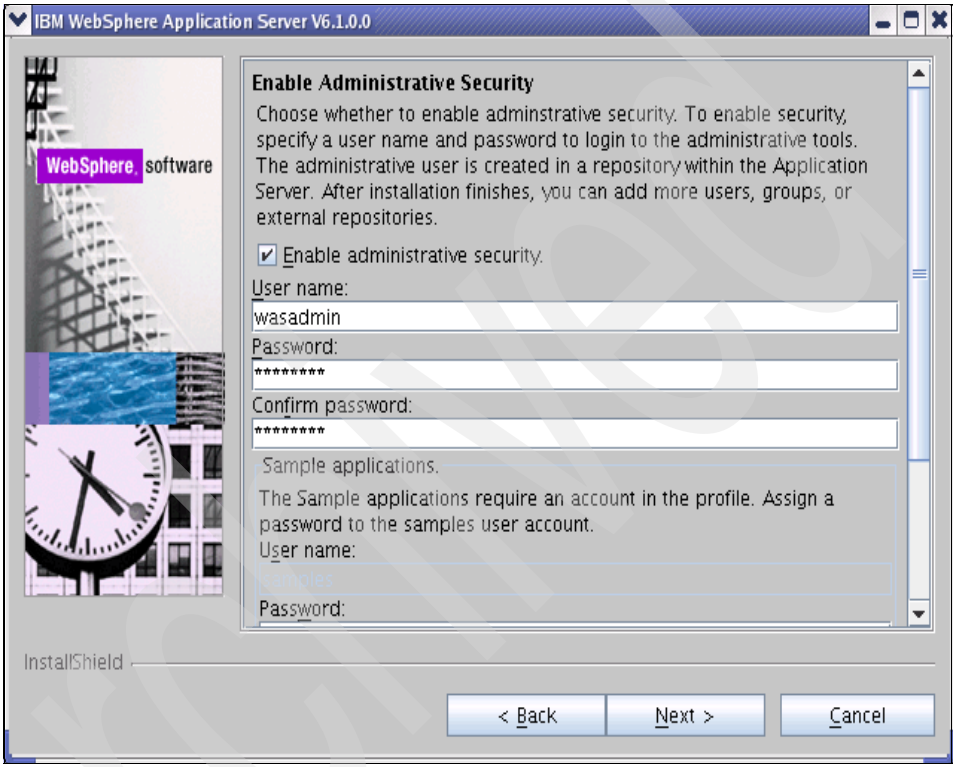


Figure 8-2 Enable Administrative Security

With administrative security enabled, the user name above must be added with the -user option and the password above must be added with the -password option to get the three command-line scripts startServer.sh, stopServer.sh, and wsadmin.sh to run successfully.

Neither Application nor Java 2 security was enabled, but with administrative security enabled, the user LocalOSUserID had to be added under both Users and Groups and Bus Security, as shown in the following WebSphere Application Server administrator windows. This enables the JMS API used by Trade 6 to work correctly. The default password for this ID is password and is also needed. If administrative security is not enabled, then adding this ID in the following steps is not necessary.

Add LocalOSUser under Users and Groups

Expand **Users and Groups** on the left hand side of the administrative console, select **Manage Users**, and then select **Create** in the window that appears (shown in Figure 8-3).

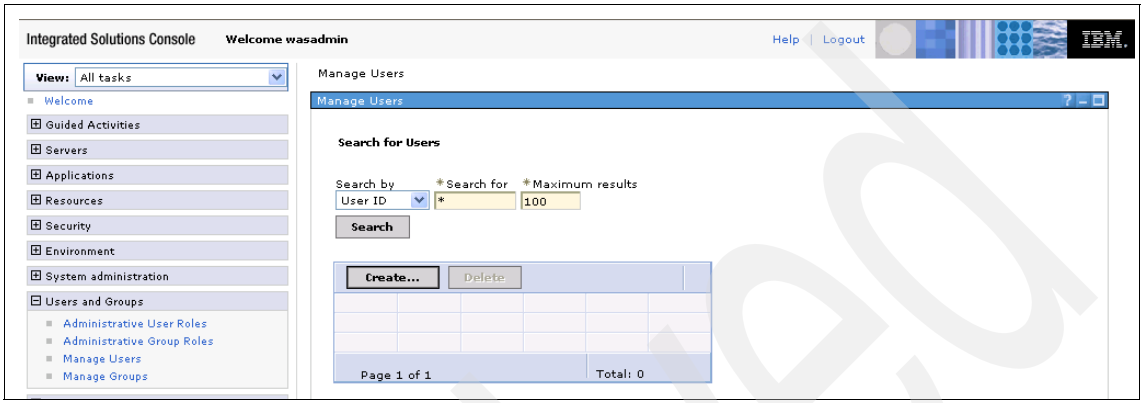


Figure 8-3 Create User under Users and Groups

The window shown in Figure 8-4 appears and the “LocalOSUserID” and “password” can be entered. After the fields are filled in, select **OK** to add the LocalOSUerID user.

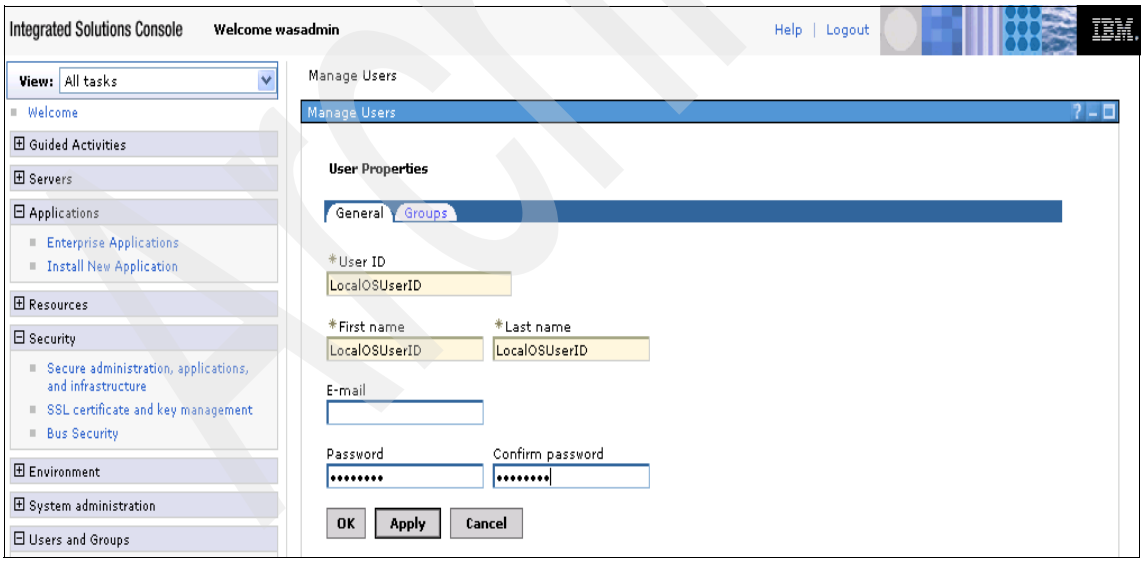


Figure 8-4 Add LocalOSUserID for JMS API

Once the message appears that the ID has been created successfully, select **Close** and the window shown in Figure 8-5 appears. This window shows that the LocalOSUserID has been added.

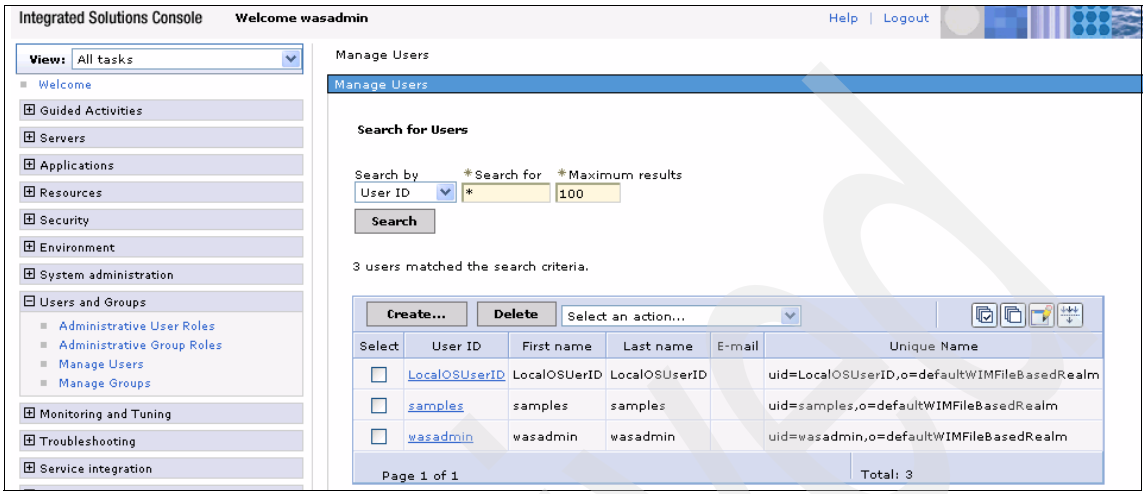


Figure 8-5 Users Summary under Users and Groups

If nothing appears in the window, select the **Search** option to populate the window with the current user IDs.

Add LocalOSUserID to Bus Security

Expand the **Security** option on the left hand side of the administrative console select Bus Security. The window shown in Figure 8-6 on page 157 appears.

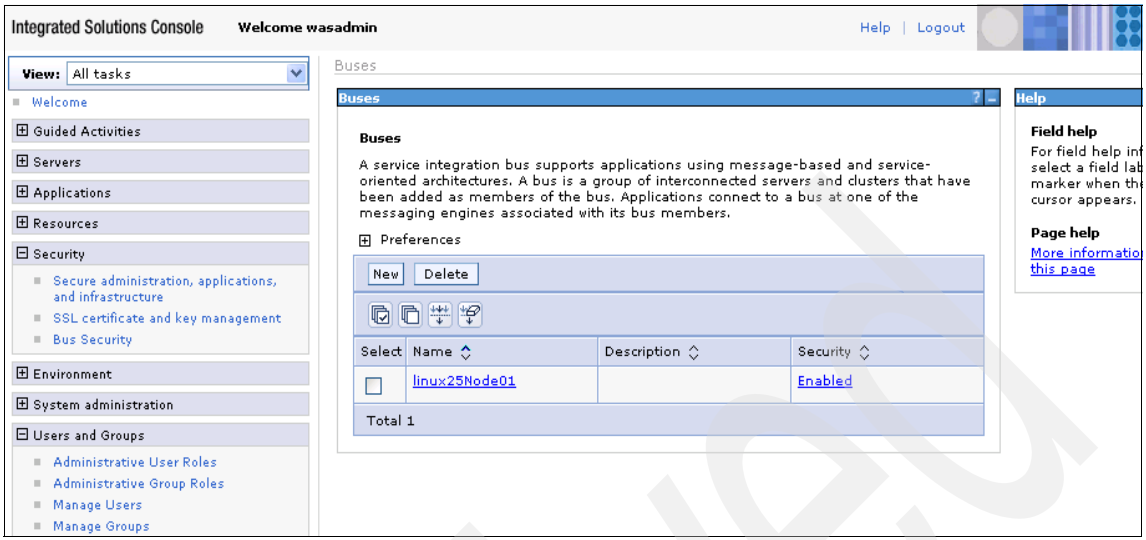


Figure 8-6 Initial Bus Security window

Click **Enabled**. The window shown in Figure 8-7 appears.

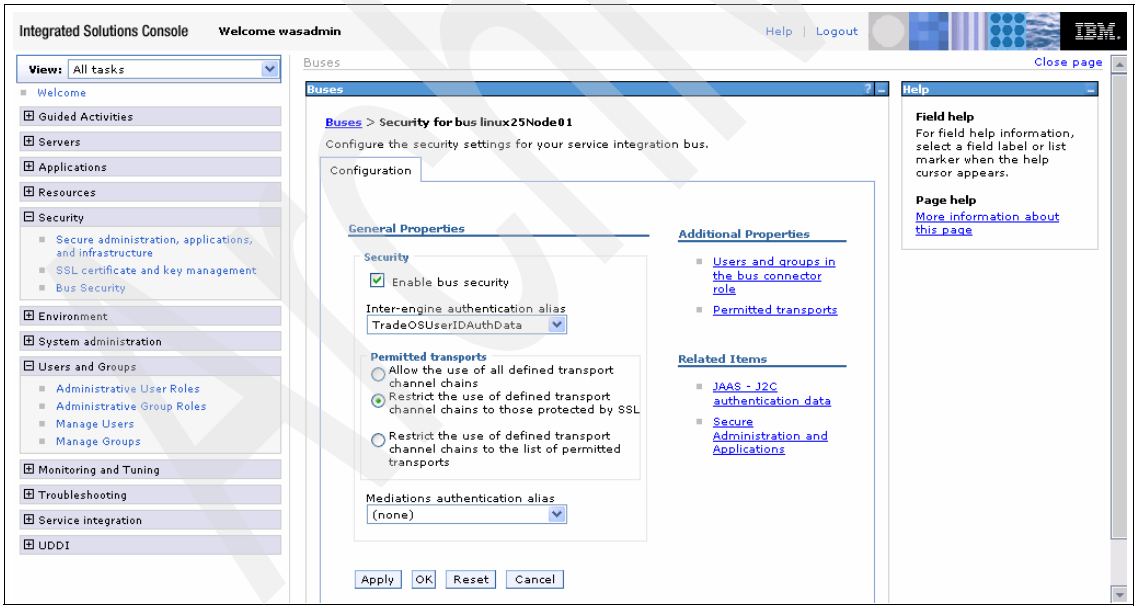


Figure 8-7 Bus Security configuration

Under Additional Properties, select the **Users and Groups** in the bus connector role. The window shown in Figure 8-8 appears.

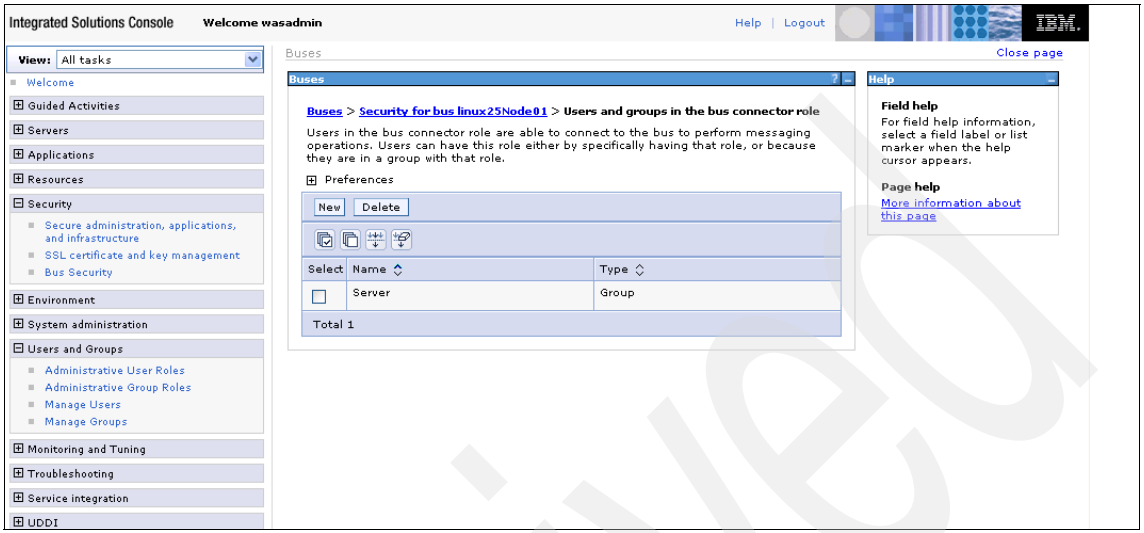


Figure 8-8 Users and Groups in the Bus Connector Role

Select **New**. The window shown in Figure 8-9 appears, where the user ID can be entered.

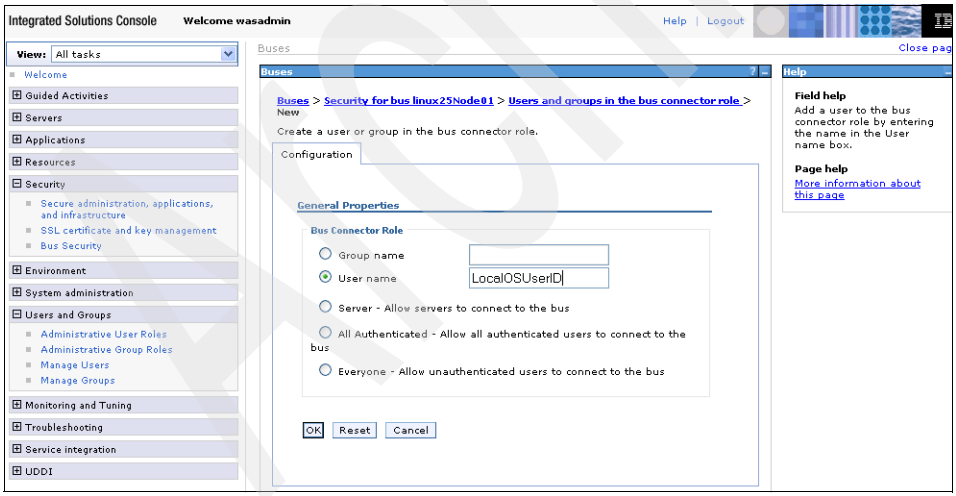


Figure 8-9 Add LocalOSUserID to the Bus Connector Role

After the user ID LocalOSUserID is entered, select **OK**, and the window shown in Figure 8-10 on page 159 will appear.

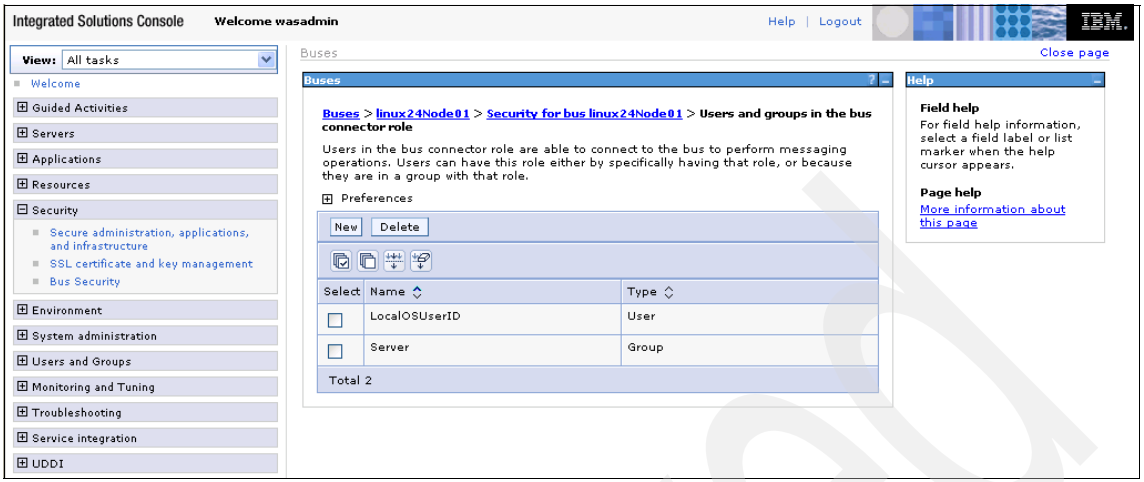


Figure 8-10 Users and Groups in Bus Connector Role

When administrative security was not enabled in WebSphere Application Server, the Trade 6 application started and ran successfully without making the two changes shown above for the LocalOSUserID user ID. After making the changes, the window shown in Figure 8-11 appears and the changes need to be saved. At this point, the Trade 6 application started successfully without stopping and restarting the WebSphere Application Server. For other configuration changes, the server may need to be stopped and restarted with the stopServer.sh and startServer.sh scripts.

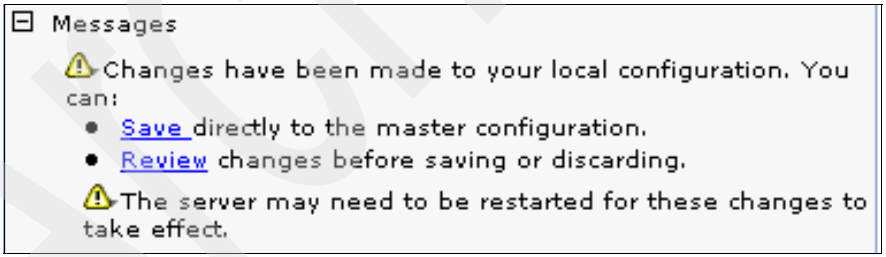


Figure 8-11 Message to save configuration changes

Figure 8-12 shows the installation summary window for the IBM WebSphere Application Server V6.1.0.0. The default name for the application server is server1.

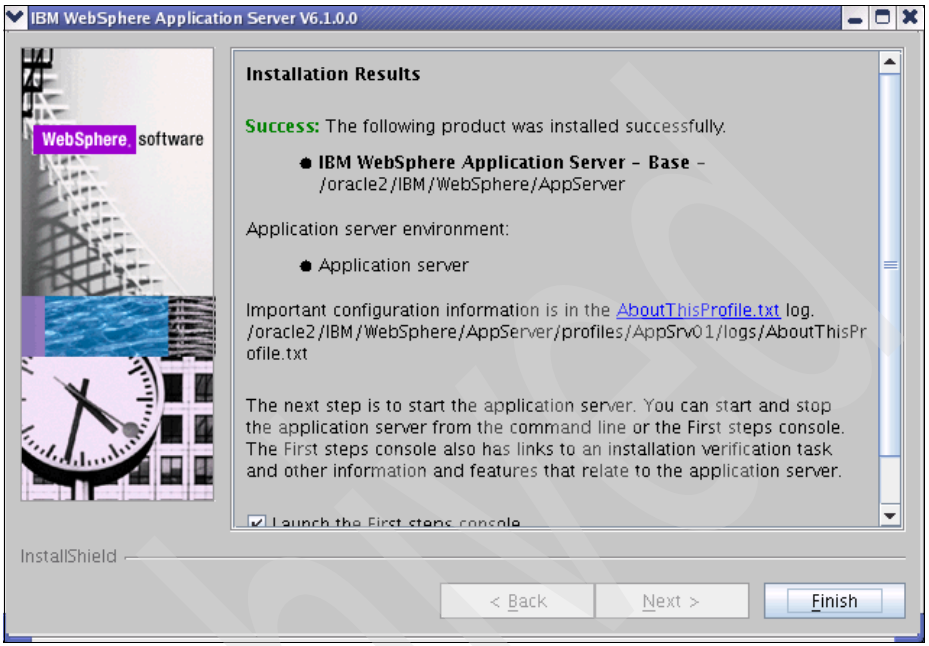


Figure 8-12 WebSphere Application Server install results

From here, the First steps console, shown in Figure 8-13 on page 161, can be accessed to start the application server and run the Installation Verifications Tests.

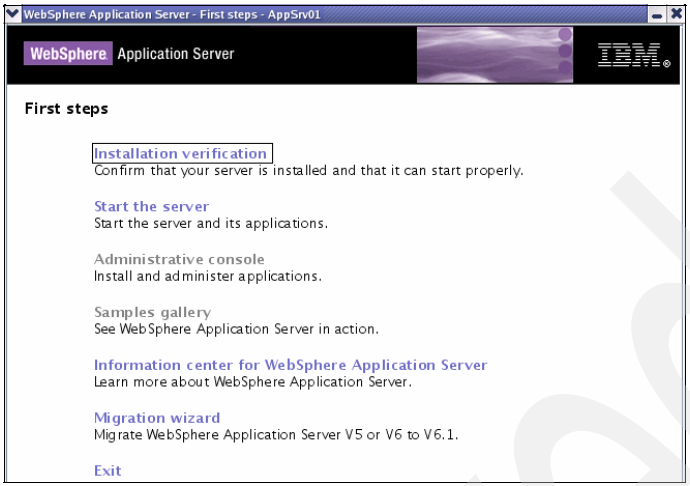


Figure 8-13 WebSphere Application Server first steps to get started

The launchpad console can also be used to install the IBM HTTP Server V6.1.0.0.

Figure 8-14 shows the install summary window for the HTTP server. The default name for the Web server is webserver1.

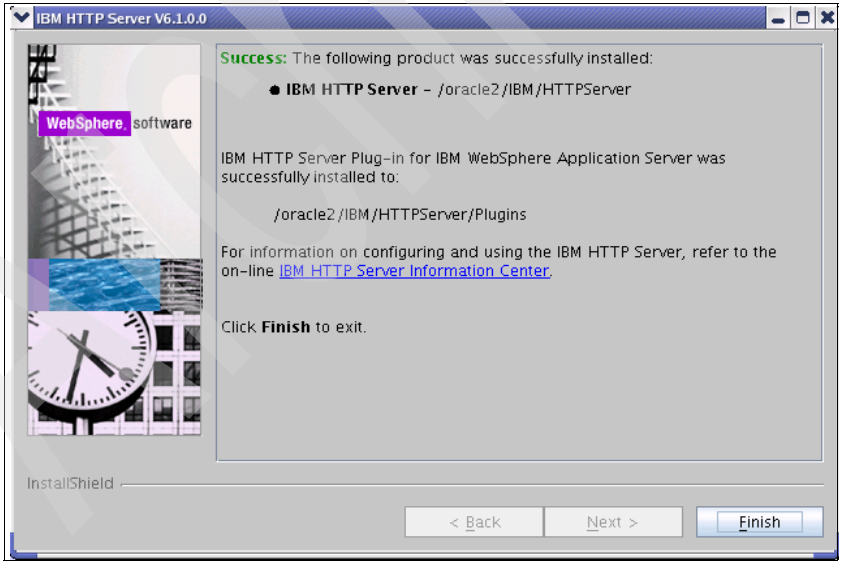


Figure 8-14 HTTP server install results

8.3 Download IBM Trade 6 Performance Benchmark Application

The first step is to download and unzip the Trade 6 zip file, tradeInstall.zip. The download is freely available from IBM at:

https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=trade6

tradeinstall is the high level directory in the downloaded zip file. An overview and architectural description of the Trade 6 application is in the PDF located in the tradeinstall/tradeTech.pdf file, and the installation instructions are in the readme file located in tradeinstall/Readme.html. The DDL to create the tables and indexes used by Trade 6 is located in tradeinstall/Oracle/Table.ddl.

8.3.1 Set up an existing Oracle Database for Trade 6

An existing Oracle database (SID = psft1) on linux25 was used for the Trade 6 tables and indexes. A new data table space and a new temporary table space were created for the Trade 6 application with the DDL shown in Example 8-1.

Example 8-1 Create Tablespace SQL statements

```
CREATE TABLESPACE TRADE61_DEFAULT_TS DATAFILE
'/oracle2/trade61/oradata/psft1/trade61dts.dbf' SIZE 400M EXTENT
MANAGEMENT LOCAL UNIFORM SIZE 10M SEGMENT SPACE MANAGEMENT AUTO;

CREATE temporary TABLESPACE TRADE61_TEMP_TS tempFILE
'/oracle2/trade61/oradata/psft1/trade61tts.dbf' SIZE 100M EXTENT
MANAGEMENT LOCAL UNIFORM SIZE 10M;

ALTER DATABASE DATAFILE '/oracle2/trade61/oradata/psft1/trade61dts.dbf'
AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;

ALTER DATABASE tempFILE '/oracle2/trade61/oradata/psft1/trade61tts.dbf'
AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;
```

The trade user name was created and authorizations were given to the trade user ID in the existing Oracle database (see Example 8-2).

Example 8-2 Create example

```
create user trade identified by trade default tablespace
TRADE61_DEFAULT_TS
temporary tablespace TRADE61_TEMP_TS account unlock;
```

```
GRANT CREATE SESSION to trade;

GRANT CREATE table to trade;
alter user trade quota unlimited on trade61_default_ts;
```

A connection was made to the Oracle database with the trade user name and the password shown here:

```
connect trade/trade@psft1
```

The Trade 6 tables and indexes were created in the trade schema in table space TRADE61_DEFAULT_TS by running the DDL:

```
tradeinstall/Oracle/Table.ddl
```

8.4 Install Trade 6 as a WebSphere Application Server application

According to the installation readme document, the tradeinstall directory was made the current directory and the WebSphere Application Server script, **wsadmin.sh -f trade.jacl**, was run from the linux25 command line to interactively install the JDBC/JMS resources and Trade 6 application:

```
[root@linux25 tradeinstall]#
/oracle2/IBM/WebSphere/AppServer/bin/wsadmin.sh -f trade.jacl
 -user wasadmin -password wasadmin
```

Since administration security was enabled, the appropriate admin user name and password had to added as options so that the WebSphere Application Server command would execute successfully. The script ran interactively, and here are some samples of the interaction:

```
Select the backend database type (db2|oracle|db2zos) [db2]:
oracle

Please enter the database driver path
[c:/oracle/product/10.1.0/db_1/jdbc/lib/ojdbc14.jar]:
/oracle1/db1/jdbc/lib/ojdbc14.jar

Please enter the Oracle database SID [tradedb]:
psft1
```

Once the script completed, the WebSphere Application Server was stopped and restarted:

```
[root@linux25 bin]# ./stopServer.sh server1 -user wasadmin -password
wasadmin
[root@linux25 bin]# ./startServer.sh server1 -user wasadmin -password
wasadmin
```

As soon as the application server was started, a browser was pointed at:

```
http://linux25:9080/trade
```

and the Trade 6 application was accessed (see Figure 8-15).

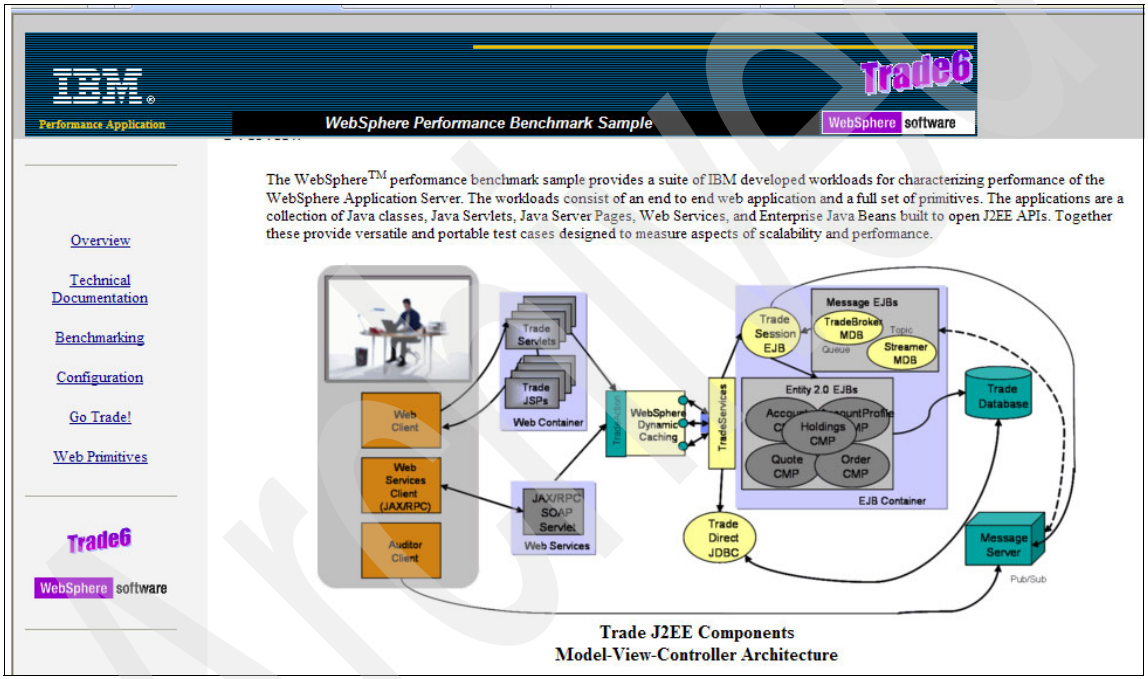


Figure 8-15 Browser access to Trade 6 application

The configuration option was selected to populate the Oracle database with the Trade 6 data. The action presents the configuration utilities window, where you must take further action to populate the Oracle database with the Trade 6 data (see Figure 8-16 on page 165).

8.5 Populate Trade 6 Oracle 10.2.0.2 database

Select **(Re)-populate Trade Database** to load the Trade 6 Oracle database with Trade 6 data, as shown in Figure 8-17.

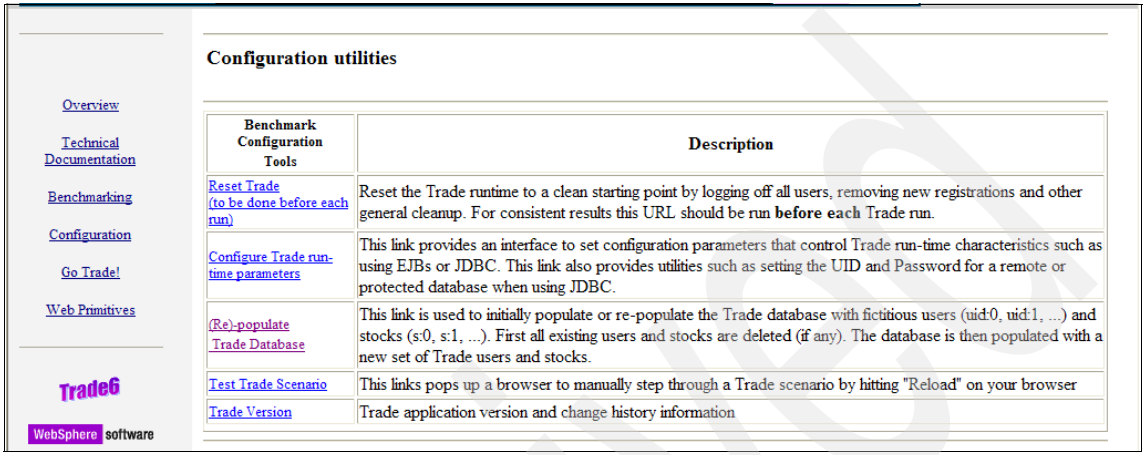


Figure 8-16 Trade 6 configuration utilities

Figure 8-17 shows the process of the loading of the Oracle Database.

```
TradeBuildDB: Building Trade Database...
This operation will take several minutes. Please wait...
TradeBuildDB: **** Creating 1000 Quotes ****
.....s:0 -
.....s:10.....s:20.....s:30.....s:40.....s:50.....s:60.....s:70.....s:80.....s:90.....s:100 -
.....s:110.....s:120.....s:130.....s:140.....s:150.....s:160.....s:170.....s:180.....s:190.....s:200 -
.....s:210.....s:220.....s:230.....s:240.....s:250.....s:260.....s:270.....s:280.....s:290.....s:300 -
.....s:310.....s:320.....s:330.....s:340.....s:350.....s:360.....s:370.....s:380.....s:390.....s:400 -
.....s:410.....s:420.....s:430.....s:440.....s:450.....s:460.....s:470.....s:480.....s:490.....s:500 -
.....s:510.....s:520.....s:530.....s:540.....s:550.....s:560.....s:570.....s:580.....s:590.....s:600 -
.....s:610.....s:620.....s:630.....s:640.....s:650.....s:660.....s:670.....s:680.....s:690.....s:700 -
.....s:710.....s:720.....s:730.....s:740.....s:750.....s:760.....s:770.....s:780.....s:790.....s:800 -
.....s:810.....s:820.....s:830.....s:840.....s:850.....s:860.....s:870.....s:880.....s:890.....s:900 -
.....s:910.....s:920.....s:930.....s:940.....s:950.....s:960.....s:970.....s:980.....s:990

**** Registering 500 Users ****
Account# 2000 userID=uid:0 has 7 holdings.
Account# 2050 userID=uid:50 has 5 holdings.
```

Figure 8-17 Load Trade 6 Oracle Database

Example 8-3 shows the data loaded by the Trade 6 application with the default settings.

Example 8-3 Output of the select statement

```
select table_name,num_rows,avg_row_len,(num_rows * avg_row_len)
size_in_bytes from user_tables where
tablespace_name='TRADE61_DEFAULT_TS';

TABLE_NAME           NUM_ROWS     AVG_ROW_LEN   SIZE_IN_BYTES
-----------------    ---------    -------------  ------------
HOLDINGEJB             2487          32            79584
ACCOUNTPROFILEEJB       500          76            38000
QUOTEEJB               1000          55            55000
KEYGENEJB                 0           0                0
ACCOUNTEJB              500          45            22500
ORDEREJB               2487          59           146733
```

Browser access to Trade 6

Trade 6 has been successfully installed and the different functions of Trade 6 can be accessed in the window shown in Figure 8-18.

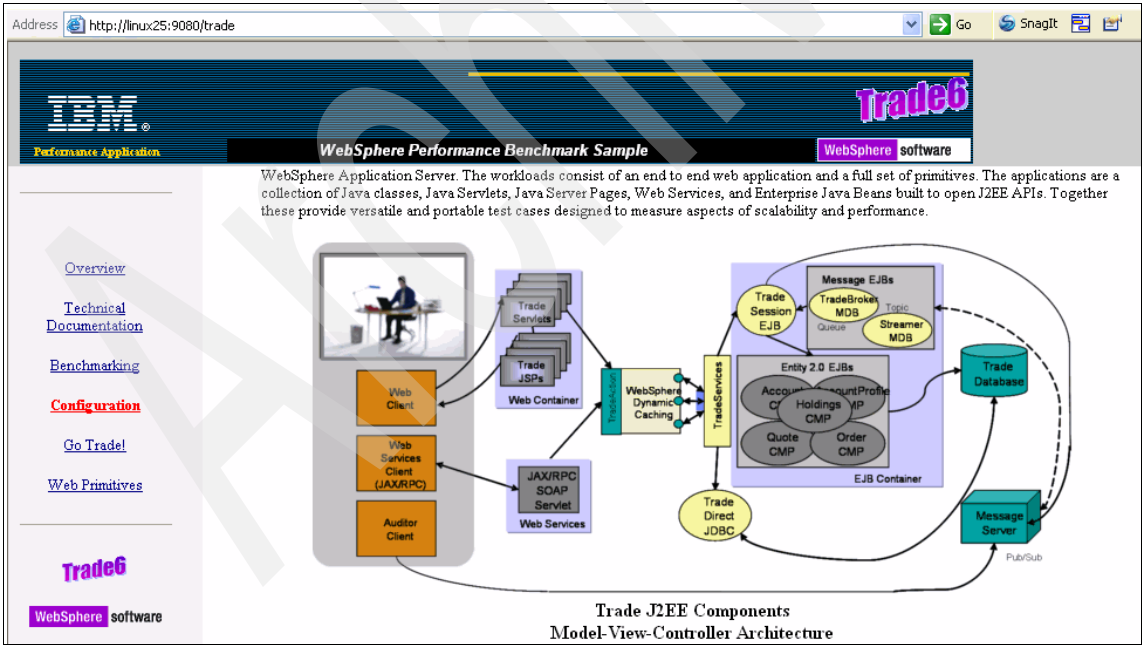


Figure 8-18 Browser access to Trade 6

9

Using Oracle Data Guard

Oracle Data Guard is an Oracle Database feature that can be used to enhance high availability.

This chapter describes an exercise we did to create a Oracle Data Guard setup with 10gR2 10.2.0.3 on Linux for System z.

The main steps were:

- Create a production database.
- Create a standby database.
- Test a switchover.

This chapter is not meant to cover the detailed installation process for Oracle products. The installation of a 10.2.0.2 database is described in Chapter 3, “Installing an Oracle 10gR2 single instance” on page 19.

9.1 Oracle Data Guard overview

Oracle Data Guard is the management, monitoring, and automation software infrastructure that creates, maintains, and monitors one or more standby databases to protect enterprise data from failures, disasters, errors, and corruptions.

Oracle Data Guard maintains these standby databases as synchronized copies of the production database. These standby databases can be located at remote disaster recovery sites thousands of miles away from the production data center, or they may be located in the same city, same campus, or even in the same building. If the production database becomes unavailable because of a planned or an unplanned outage, Oracle Data Guard can switch any standby database to the production role, thus minimizing the downtime associated with the outage, and preventing any data loss.

A standby database is initially created from a backup copy of the primary database. Once created, Oracle Data Guard automatically maintains the standby database as a synchronized copy of the primary database by transmitting primary database redo data to the standby system and then applying the redo data to the standby database.

A physical standby database provides a physically identical copy of the primary database, with on-disk database structures that are identical to the primary database on a block-for-block basis. The database schema, including indexes, is the same.

A logical standby database contains the same logical information as the production database, although the physical organization and structure of the data can be different.

Using Oracle Data Guard, the role of a database can be switched from a primary role to a standby role and vice versa, ensuring no data loss in the process, and minimizing downtime. There are two kinds of role transitions: a switchover and a failover.

9.2 Oracle Data Guard Broker

The Oracle Data Guard Broker is a distributed management framework that automates and centralizes the creation, maintenance, and monitoring of Oracle Data Guard configurations. All management operations can be performed either through Oracle Enterprise Manager, which uses the Broker, or through the Broker's specialized command-line interface (DGMGRL). Oracle Data Guard

Broker 11g also enables automatic database failover for Oracle Data Guard configurations using either Maximum Availability or Maximum Performance modes.

9.3 Oracle Data Guard benefits

1. Disaster recovery and high availability

   Oracle Data Guard provides an efficient and comprehensive disaster recovery and high availability solution. Automatic failover and easy-to-manage switchover capabilities allow quick role reversals between primary and standby databases, minimizing the downtime of the primary database for planned and unplanned outages.

2. Complete data protection

   A standby database also provides an effective safeguard against data corruption and user errors. Storage level physical corruption on the primary database does not propagate to the standby database. Similarly, logical corruptions or user errors that cause the primary database to be permanently damaged can be resolved. Finally, the redo data is validated at the time it is received at the standby database and further when applied to the standby database.

3. Efficient utilization of system resources

   A physical standby database can be used for backups and read-only reporting, thereby reducing the primary database workload and saving valuable CPU and I/O cycles. A physical standby database can also be easily converted back and forth between being a physical standby database and an open read/write database, without compromising data protection. A logical standby database enables read-write access to a synchronized standby database, adding local tables to the standby database that can also be updated, or creating additional indexes to optimize read performance.

4. Flexibility in data protection to balance availability against performance requirements

   Oracle Data Guard offers the maximum protection, maximum availability, and maximum performance modes to help enterprises balance data availability against system performance requirements.

5. Protection from communication failures

   If network connectivity is lost between the primary and one or more standby databases, redo data cannot be sent from the primary to those standby databases. Once connectivity is re-established, the missing redo data is automatically detected by Oracle Data Guard and the necessary archive logs are automatically transmitted to the standby databases. The standby databases are resynchronized with the primary database, with no manual intervention by the administrator.

6. Centralized and simple management

   Oracle Data Guard Broker automates the management and monitoring tasks across the multiple databases in a Oracle Data Guard configuration. Administrators may use either Oracle Enterprise Manager or the Broker's own specialized command-line interface (DGMGRL) to take advantage of this integrated management framework.

7. Integrated with Oracle database

   Oracle Data Guard is available as an integrated feature of the Oracle Database (Enterprise Edition) at no extra cost.

9.4 Steps to creating an Oracle Data Guard solution

The steps we executed were:

- Install the Oracle Binaries.
- Create a database on a production node by using DBCA.
- Prepare a production database for copy.
- Copy a database to standby node.
- Set up the standby database.
- Start the physical standby database.
- Test the switchover process.

The documentation we followed was:

- *Oracle Data Guard Concepts and Administration 10g Release 2 (10.2)*, B14239-04
- *MetaLink Note: 232240.1 Performing switchover in Oracle Data Guard Configuration*
- *Oracle Database 10g High Availability with RAC, Flashback & Oracle Data Guard*

The primary (production) and standby databases can be in the same Linux guest, in two Linux guests on the same LPAR, in two Linux guests, each on its own LPAR, on the same z9, or on two LPARS on separate z9s. Our structure was two Linux guests in separate LPARs in the same z9, as shown in Figure 9-1.

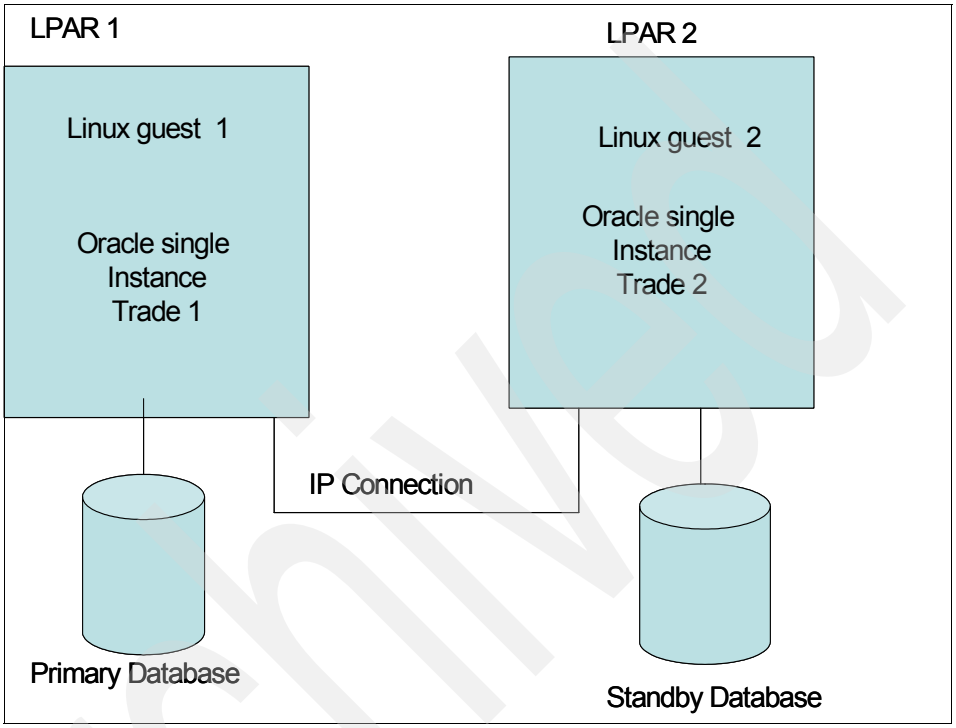


Figure 9-1 Oracle Data Guard setup

9.5 Installation and creation of production database

This chapter does not cover the details of the installation of Oracle binaries or the creation of a database, as this is covered in other books or chapters.

9.6 Prepare the production database

The first step to create a Physical Standby Database is to prepare the production database in order to be able to copy it to the standby database server.

In our example, we used two guests running SLES10 and Oracle 10gR2. The database name was tradedb, the unique name on the first node was trade1, and the unique name on the second node was trade2. The directory structure was the same for both nodes:

```
ORACLE_HOME=/oracle/10gr2
LOG_ARCHIVE_DESTINATION=/oracle/trade
ORACLE DATABASE FILE LOCATION=/oracle/10gr2/oradata/tradedb
```

These are the steps we executed to prepare the database for copying.

9.6.1 Enable forced logging

Place the primary database in FORCE LOGGING mode after database creation using the following SQL statement:

```
SQL> ALTER DATABASE FORCE LOGGING;
```

To verify that force logging is set, run:

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

FOR
---
YES
```

9.6.2 Create a password file

This was done during the installation of Oracle Binaries and the creation of the database.

9.6.3 Configure standby redo logs

A standby redo log is required for the maximum protection and maximum availability modes and the LGWR ASYNC transport mode is recommended for all databases. Oracle Data Guard can recover and apply more redo data from a standby redo log than from archived redo log files alone. To configure the logs, do the following steps:

1. Ensure that the log file sizes are identical on the primary and standby databases.
2. Determine the appropriate number of standby redo log file groups using the following formula:

   (Max num of log files per thread + 1) * (Max num of threads) = 4

3. Verify the related database parameters and settings:

```
SQL> SELECT TYPE,RECORDS_TOTAL FROM V$CONTROLFILE_RECORD_SECTION
WHERE TYPE='REDO LOG';

TYPE                         RECORDS_TOTAL
---------------------------- -------------
REDO LOG                                16
SQL> SELECT DIMLM FROM X$KCCDI;

DIMLM
----------
3
```

4. Create standby redo log file groups:

```
SQL>  ALTER DATABASE ADD STANDBY LOGFILE
   2 GROUP 4 '/oracle/10gr2/oradata/tradedb/redo04s.log' size 50M,
   3 GROUP 5 '/oracle/10gr2/oradata/tradedb/redo05s.log' size 50M,
   4 GROUP 6 '/oracle/10gr2/oradata/tradedb/redo06s.log' size 50M,
   5 GROUP 7 '/oracle/10gr2/oradata/tradedb/redo07s.log' size 50M;
```

5. Verify that the standby redo log file groups were created:

```
SQL> SELECT GROUP#,THREAD#,SEQUENCE#,ARCHIVED,STATUS FROM
V$STANDBY_LOG;

    GROUP#    THREAD#  SEQUENCE# ARC STATUS
---------- ---------- ---------- --- ----------
         4          0          0 YES UNASSIGNED
         5          0          0 YES UNASSIGNED
         6          0          0 YES UNASSIGNED
         7          0          0 YES UNASSIGNED
```

9.6.4 Setting primary database initialization parameters

There are two ways to set the initialization parameters: one is to create a init{ORACLE_SID}.ora file and start the database with that file. The second way is to issue the following commands:

1. DB_UNIQUE_NAME
2. LOG_ARCHIVE_CONFIG
3. LOG_ARCHIVE_DEST
4. ALTER SYSTEM to set flash recovery on

The following sections show how to set these commands.

DB_UNIQUE_NAME

DB_UNIQUE_NAME: Specify a unique name for each database. This name stays with the database and does not change, even if the primary and standby databases reverse roles. The syntax is:

```
SQL> SELECT * FROM V$DATAGUARD_CONFIG;
DB_UNIQUE_NAME
------------------------------
tradedb

SQL> ALTER SYSTEM SET DB_UNIQUE_NAME='trade1' SCOPE=SPFILE;
SQL> SELECT * FROM V$DATAGUARD_CONFIG;
DB_UNIQUE_NAME
------------------------------
tradedb
```

At this point, we need to shut down the database and restart it in order to see the changes:

```
SQL> shutdown immediate
Database dismounted.
ORACLE instance shut down.

SQL> startup
ORACLE instance started.

Total System Global Area 1241513984 bytes
Fixed Size                  2072384 bytes
Variable Size             318767296 bytes
Database Buffers          905969664 bytes
Redo Buffers               14704640 bytes
Database mounted.
Database opened.

SQL> SELECT * FROM V$DATAGUARD_CONFIG;

DB_UNIQUE_NAME
------------------------------
trade1
```

LOG_ARCHIVE_CONFIG

LOG_ARCHIVE_CONFIG: Specify the DG_CONFIG attribute on this parameter to list the DB_UNIQUE_NAME of the primary and standby databases in the Oracle Data Guard configuration. This enables the dynamic addition of a standby database to a Oracle Data Guard configuration that has a Real Application

Clusters primary database running in either maximum protection or maximum availability mode. By default, the LOG_ARCHIVE_CONFIG parameter enables the database to send and receive redo. After a role transition, you may need to specify these settings again using the SEND, NOSEND, RECEIVE, or NORECEIVE keywords. Use the following syntax:

```
set DB_UNIQUE_NAME of primary DB and standby DB.

SQL> ALTER SYSTEM SET LOG_ARCHIVE_CONFIG='DG_CONFIG=(trade1,trade1)';

SQL> SHOW PARAMETER LOG_ARCHIVE_CONFIG

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
LOG_ARCHIVE_CONFIG                   STRING
DG_CONFIG=(trade1,trade2)
SQL> SELECT * FROM V$DATAGUARD_CONFIG;

DB_UNIQUE_NAME
------------------------------
trade1
trade2
```

LOG_ARCHIVE_DEST

Set up the ARCHIVE LOG file (where the ARCHIVE LOG file output is):

- Regardless of the role (primary or standby), output the ARCHIVE LOG to /oracle/trade on local according to log_archive_dest_1.
- Only in the case of the primary role should you output ARCHIVE LOG to orcl_2 on remote also according to log_archive_dest_2.

LOG_ARCHIVE_DEST_n: Specify where the redo data is to be archived on the primary and standby systems.

LOG_ARCHIVE_DEST_1: Archives redo data generated by the primary database from the local online redo log files to the local archived redo log files in /oracle/trade/.

LOG_ARCHIVE_DEST_2: Valid only for the primary role. This destination transmits redo data to the remote physical standby destination trade2.

Example 9-1 shows a LOG _ARCHIVE_DEST example.

Example 9-1 LOG _ARCHIVE_DEST example

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='LOCATION=/oracle/trade
  2  VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
  3  DB_UNIQUE_NAME=trade1';
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_2='SERVICE=trade2
  2  VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
  3  DB_UNIQUE_NAME=trade2';

SQL> SHOW PARAMETER LOG_ARCHIVE_DEST_1

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
log_archive_dest_1                   string      LOCATION=/oracle/trade VALID_F
                                                 OR=(ALL_LOGFILES,ALL_ROLES) DB
                                                 _UNIQUE_NAME=trade1
log_archive_dest_10                  string

SQL> SHOW PARAMETER LOG_ARCHIVE_DEST_2;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
log_archive_dest_2                   string      SERVICE=trade2 VALID_FOR=(ONLI
                                                 NE_LOGFILES,PRIMARY_ROLE) DB_U
                                                 NIQUE_NAME=trade2

SQL> SELECT DEST_ID,TARGET,DESTINATION,VALID_TYPE,VALID_ROLE,DB_UNIQUE_NAME FROM
V$ARCHIVE_DEST WHERE DEST_ID<=2;

DEST_ID TARGET
---------- -------
DESTINATION
--------------------------------------------------------------------------------
VALID_TYPE      VALID_ROLE   DB_UNIQUE_NAME
--------------- ------------ ------------------------------
1 PRIMARY
/oracle/trade/
ALL_LOGFILES    ALL_ROLES    trade2

2 STANDBY
trade1
ONLINE_LOGFILE  PRIMARY_ROLE trade1
```

ALTER SYSTEM to set the flash recovery on

On the physical standby database that will be activated for read/write access, you should set the following initialization parameters to ensure a guaranteed restore point can be created. This sets up the flash recovery area.

```
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE=2G;


SQL> ALTER SYSTEM SET
DB_RECOVERY_FILE_DEST='/oracle/10gr2/flash_recovery_area/TRADEDB';


SQL> SHOW PARAMETER DB_RECOVERY_FILE_DEST

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
db_recovery_file_dest                string
/oracle/10gr2/flash_recovery_a
                                                 rea/TRADEDB
db_recovery_file_dest_size           big integer 2G


SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP MOUNT
SQL> ALTER DATABASE FLASHBACK ON;


SQL> ALTER DATABASE OPEN;
```

At this point you should have all the initialization parameters set. You can create a pfile as follows:

```
SQL>CREATE PFILE='/tmp/tradedb_node1.ora' FROM SPFILE;
```

Here are the parameters in the pfile that are related to the Oracle Data Guard:

```
DB_NAME=tradedb
DB_UNIQUE_NAME=trade1
SERVICE_NAMES=trade1
LOG_ARCHIVE_CONFIG='DG_CONFIG=(trade1,trade2)'
LOG_ARCHIVE_DEST_1=
 'LOCATION=/oracle/trade/
  VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
  DB_UNIQUE_NAME=trade1'
LOG_ARCHIVE_DEST_2=
 'SERVICE=trade2
```

```
  VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
  DB_UNIQUE_NAME=trade2'
LOG_ARCHIVE_DEST_STATE_1=ENABLE
LOG_ARCHIVE_DEST_STATE_2=ENABLE
REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE
LOG_ARCHIVE_FORMAT=%t_%s_%r.arc
LOG_ARCHIVE_MAX_PROCESSES=30
```

Pfile from production database

This is the pfile used for the production database:

Example 9-2 pfile for the production database

```
/tmp/tradedb_node1.ora'

tradedb.__db_cache_size=905969664
tradedb.__java_pool_size=16777216
tradedb.__large_pool_size=16777216
tradedb.__shared_pool_size=285212672
tradedb.__streams_pool_size=0
*.audit_file_dest='/oracle/10gr2/admin/tradedb/adump'
*.background_dump_dest='/oracle/10gr2/admin/tradedb/bdump'
*.compatible='10.2.0.3.0'
*.control_files='/oracle/10gr2/oradata/tradedb/control01.ctl','/oracle/10gr2/oradata/
tradedb/control02.ctl','/oracle/10gr2/oradata/tradedb/control03.ctl'
*.core_dump_dest='/oracle/10gr2/admin/tradedb/cdump'
*.db_block_size=8192
*.db_domain=''
*.db_file_multiblock_read_count=16
*.db_name='tradedb'
*.db_recovery_file_dest_size=2147483648
*.db_recovery_file_dest='/oracle/10gr2/flash_recovery_area/TRADEDB'
*.db_unique_name='trade1'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=tradedbXDB)'
*.fal_client='TRADE1'
*.fal_server='TRADE2'
*.job_queue_processes=10
*.log_archive_config='DG_CONFIG=(trade1,trade2)'
*.log_archive_dest_1='LOCATION=/oracle/trade VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=trade1'
*.log_archive_dest_2='SERVICE=trade2 VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=trade2'
*.LOG_ARCHIVE_DEST_STATE_1='ENABLE'
*.LOG_ARCHIVE_DEST_STATE_2='ENABLE'
*.LOG_ARCHIVE_FORMAT='%t_%s_%r.arc'
```

```
*.LOG_ARCHIVE_MAX_PROCESSES=30
*.open_cursors=300
*.pga_aggregate_target=408944640
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=1228931072
*.standby_file_management='AUTO'
*.undo_management='AUTO'
*.undo_tablespace='UNDOTBS1'
*.user_dump_dest='/oracle/10gr2/admin/tradedb/udump'
```

Enable archiving

At this point, we want to make sure that archiving is enabled, so we use the following commands:

```
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
SQL> ALTER DATABASE ARCHIVELOG;
SQL> ALTER DATABASE OPEN;
```

9.7 Creating a physical standby database

In order to use Oracle Data Guard, you need a copy of the database. There are several ways to create a backup of a database. Oracle recommends RMAN, which is a tool integrated with the Oracle Database that satisfies the demands of high performance, manageable backup, and recovery. Another way is to shut down the database and copy the DBF files. In our example, we used this method.

9.7.1 On primary

- Create a backup copy of the primary database datafiles.

  In order to set up a Standby Database, we will need to create a copy of the primary database. This can be accomplished by shutting down the database and copying the dbf files from the primary node to the standby node. We used the **scp** command.

- Create a control file for the standby database.

  We need a control file for the standby node, which is done on the primary node with the following command:

  ```
  SQL> ALTER DATABASE CREATE STANDBY CONTROLFILE AS '/tmp/trade.ctl';
  ```

- Prepare an Initialization Parameter File for the standby database.

  The Initialization Parameter File for the standby node needs to be modified to accommodate the standby node environment, as shown in Example 9-3.

Pfile for standby database

Example 9-3 shows the pfile we used. Note the lines in bold.

Example 9-3 pfile for standby database

```
/tmp/tradedb_node2.ora'

tradedb.__db_cache_size=905969664
tradedb.__java_pool_size=16777216
tradedb.__large_pool_size=16777216
tradedb.__shared_pool_size=285212672
tradedb.__streams_pool_size=0
*.audit_file_dest='/oracle/10gr2/admin/tradedb/adump'
*.background_dump_dest='/oracle/10gr2/admin/tradedb/bdump'
*.compatible='10.2.0.3.0'
*.control_files='/oracle/10gr2/oradata/tradedb/control01.ctl','/oracle/10gr2/oradata/
tradedb/control02.ctl','/oracle/10gr2/oradata/tradedb/control03.ctl'
*.core_dump_dest='/oracle/10gr2/admin/tradedb/cdump'
*.db_block_size=8192
*.db_domain=''
*.db_file_multiblock_read_count=16
*.db_name='tradedb'
*.db_recovery_file_dest='/oracle/10gr2/flash_recovery_area'
*.db_recovery_file_dest_size=2147483648
*.DB_UNIQUE_NAME='trade2'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=tradedbXDB)'
*.FAL_CLIENT='trade2'
*.FAL_SERVER='trade1'
*.job_queue_processes=10
*.LOG_ARCHIVE_CONFIG='DG_CONFIG=(trade1,trade2)'
*.LOG_ARCHIVE_DEST_1='LOCATION=/oracle/trade/ VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=trade2'
*.LOG_ARCHIVE_DEST_2='SERVICE=trade1 LGWR ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=trade1'
*.LOG_ARCHIVE_DEST_STATE_1='ENABLE'
```

```
*.LOG_ARCHIVE_DEST_STATE_2='ENABLE'
*.LOG_ARCHIVE_FORMAT='%t_%s_%r.arc'
*.LOG_ARCHIVE_MAX_PROCESSES=30
*.open_cursors=300
*.pga_aggregate_target=408944640
*.processes=150
*.REMOTE_LOGIN_PASSWORDFILE='EXCLUSIVE'
*.sga_target=1228931072
*.STANDBY_FILE_MANAGEMENT='AUTO'
*.undo_management='AUTO'
*.undo_tablespace='UNDOTBS1'
*.user_dump_dest='/oracle/10gr2/admin/tradedb/udump'
```

9.7.2 Copy files from the primary system to the standby system

These are the files we copied:

- redo01.log
- redo02.log
- redo03.log
- redo04s.log
- redo04s.log
- redo05s.log
- redo06s.log
- redo07s.log
- sysaux01.dbf
- system01.dbf
- temp01.dbf
- undotbs01.dbf
- users01.dbf

Then copy the created control file /tmp/trade.ctl to the standby node and rename the files on the standby node as follows:

- control01.ctl
- control02.ctl
- control03.ctl

9.8 Set up the standby database

In order to have the standby database synchronized with the primary database, you have to do some setup. This setup will enable the correct communication between the databases:

- Set up the environment to support the standby database.
- Create a password file.

This was done during the installation of the Oracle Binaries and the creation of the database.

Configure the listeners for the primary and standby databases. Either use **netca** or update the listner.ora files as follows:

- On primary

```
TRADE1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = node1)(PORT = 1521))
    )
  (CONNECT_DATA =
    (SERVICE_NAME = trade1)
  )
)
```

- On standby

```
TRADE2 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = node2)(PORT = 1521))
    )
  (CONNECT_DATA =
    (SERVICE_NAME = trade2)
  )
)
```

9.8.1 Create Oracle Net service names

On both the primary and standby systems, use Oracle Net Manager to create a network service name for the primary and standby databases that will be used by redo transport services.

The Oracle Net service name must resolve to a connect descriptor that uses the same protocol, host address, port, and service that you specified when you configured the listeners for the primary and standby databases. The connect descriptor must also specify that a dedicated server be used.

9.9 Starting the physical standby database

The final steps are to start the standby database. In order to start the database, we need the following:

- Create a server parameter file for the standby database:

  ```
  SQL> CREATE SPFILE FROM PFILE='inittradedb_node2.ora';
  ```

- Start the physical standby database:

  ```
  SQL> STARTUP MOUNT;
  ```

- Start redo apply:

  ```
  SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM
  SESSION;
  ```

- Verify the physical standby database is performing properly.

In order to verify that the standby database was updated, we created the tables and users needed by the application and then we started the application and populated the database. After doing a switchover, we check for tables and users on the standby database and found everything there.

9.10 Switchover to the standby database

Switchover is a role reversal between the primary database and one of its standby databases. A switchover operation guarantees no data loss. This is typically done for planned maintenance of the primary system. During a switchover, the primary database transitions to a standby role and the standby database transitions to the primary role. The transition occurs without having to re-create either database.

By contrast, a failover is an irreversible transition of a standby database to the primary role. This is only done in the event of a catastrophic failure of the primary database.

Here are the steps to do a switchover:

1. Verify that it is possible to perform a switchover operation. On the primary, query the switchover_status column of v$database to verify that switchover to standby is possible:

   ```
   SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
   SWITCHOVER_STATUS
   ------------------
   TO STANDBY
   ```

   In order to perform a switchover, all sessions to the database need to be disconnected:

   ```
   SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO STANDBY WITH SESSION
   SHUTDOWN;
   ```

2. Convert the primary database to the new standby:

   ```
   SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PHYSICAL STANDBY;
   Database altered.
   ```

3. Shut down the former primary and mount it as a standby database:

   ```
   SQL> SHUTDOWN IMMEDIATE
   ORA-01507: database not mounted
   ORACLE instance shut down.
   SQL> STARTUP NOMOUNT
   ORACLE instance started.

   Total System Global Area   85020944 bytes
   Fixed Size                   454928 bytes
   Variable Size              71303168 bytes
   Database Buffers           12582912 bytes
   Redo Buffers                 679936 bytes

   SQL> ALTER DATABASE MOUNT STANDBY DATABASE;

   Database altered.
   ```

4. Defer the remote archive destination on the old primary:

   ```
   SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=DEFER;
   ```

5. Verify that the physical standby can be converted to the new primary:

   ```
   SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
   SWITCHOVER_STATUS
   ------------------
   SWITCHOVER PENDING
   ```

Note that if the status returns SESSIONS ACTIVE then you should append the session shutdown clause to the command in the next step

6. Convert the physical standby to the new primary:

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;
Database altered.
```

7. Shut down and start up the new primary:

```
SQL> SHUTDOWN IMMEDIATE
ORA-01507: database not mounted
ORACLE instance shut down.
SQL> STARTUP
ORACLE instance started.

Total System Global Area   85020944 bytes
Fixed Size                   454928 bytes
Variable Size              71303168 bytes
Database Buffers           12582912 bytes
Redo Buffers                 679936 bytes
Database mounted.
Database opened.
SQL>
```

8. Enable remote archiving on the new primary to the new standby:

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE;
```

9. Start managed recover on the new standby database:

```
SQL> RECOVER MANAGED STANDBY DATABASE DISCONNECT;
Media recovery complete.
```

At this point, we have a fully operational Oracle Data Guard.

10

Migrating an Oracle database from z/OS to Linux for System z

Oracle has always claimed that Oracle is Oracle. It doe not matter where Oracle is running, it has the same functionality. With the trend of using Linux for System z, we found that it would be helpful to show the process of migrating a database from Oracle on z/OS to Oracle on Linux for System z.

This chapter goes through the migration of an application database from Oracle on system z/OS to Oracle on Linux for System z. The z/OS database is in EBCDIC and the Linux for System z database is in ASCII.

10.1 Migrating your Oracle Database from z/OS to Linux for System z

To migrate a database, you can use many tools, such as traditional export/import, RMAN, Data Pump export/import, and some third party tools. We decided to use Oracle Data Pump, as this is a faster migration than using the traditional export/import utilities.

Transportable table spaces are not an option because of the EBCDIC to ASCII conversion that is required. This is the reason Oracle on z/OS is not a supported platform for transportable table spaces.

10.2 Using Data Pump

The design of Data Pump Export and Import results in greatly enhanced data movement performance over the original Export and Import utilities. Information about how to use the Data Pump is located in *Oracle Database Utilities*, B10825-01.

The Data Pump Export and Import utilities (invoked with the **expdp** and **impdp** commands, respectively) have a similar look and feel to the original Export (exp) and Import (imp) utilities, but they are completely separate. Dump files generated by the new Data Pump Export utility are not compatible with dump files generated by the original Export utility. Therefore, files generated by the original Export (exp) utility cannot be imported with the Data Pump Import (impdp) utility.

Oracle recommends that you use the new Data Pump Export and Import utilities because they support all Oracle Database 10g features, except for XML schemas. Original Export and Import supports the full set of Oracle database release 9.2 features.

The exercise we did was to migrate a database that resided on Oracle on z/OS to Oracle on Linux for System z. The database was used for a Web Application to a Oracle database.

We started with Oracle installation on System z/OS. Once we had Oracle installed, we created the database, user, and schema for the Web Application.

The steps we took to do the migration test were:

1. Installing Oracle 10g on z/OS
2. Creating the database and tables

3. Populating the database
4. Preparing for Data Pump Export
5. Running Data Pump Export
6. Moving the dump file to new system
7. Installing Oracle 10g on Linux for System z
8. Creating the database
9. Preparing for Data Pump Import
10. Running Data Pump Import
11. Validating the application

10.3 Installing Oracle 10g on z/OS

We installed the Oracle binaries on z/OS by following the Oracle Installation Guide and *Experiences with Oracle Database 10g on z/OS*, SG24-7055. After installing the binaries and following the creation of a database (described in Chapter 5, “Creating the Oracle Database”, of *Experiences with Oracle Database 10g on z/OS,* SG24-7055), we needed to verify that job SQLORA13 ran successfully. To be able to use Data Pump, the Java code must be initialized successfully. Check that job SQLORA13 was completed without errors.

10.4 Creating the database and tables

Data Pump export uses the DBMS_METADAT_UTIL package. This package needs XML installed and XML needs JAVA installed. Job SQLORA13 takes care of these requirements.

Example 10-1 SQLORA13

```
//SQLORA13 JOB (0000,OR),'ORACLE INSTALL 3',CLASS=A,
//        MSGCLASS=X,PRTY=15,MSGLEVEL=(1,1),NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC53
//*-------------------------------------------------------------------*
//*  COPYRIGHT (C) 2006 ORACLE.  ALL RIGHTS RESERVED.                 *
//*-------------------------------------------------------------------*
//*                                                                   *
//*  JOB DESCRIPTION:  Initialize Java VM and XML                     *
//*                                                                   *
//*-------------------------------------------------------------------*
```

```
//*
//SQLPLUS  EXEC PGM=SQLPLUS,PARM='/NOLOG',REGION=0M
//STEPLIB  DD  DISP=SHR,DSN=ORACLE.V10202.CMDLOAD
//ORA$LIB  DD  DISP=SHR,DSN=ORACLE.V10202.MESG
//*  REQUIRES //ORA@SID  DD DUMMY STATEMENT (ORACLE INSTANCE).
//ORA@ORA1 DD DUMMY
//ORA$ENV  DD  DISP=SHR,DSN=ORACLE2.V102DB.PARMLIB(ORA1ENV)
//SQLLOGIN DD DUMMY
//SYSIN    DD *
SET ECHO ON
CONNECT / as SYSDBA
@/oracle/10202/db/javavm/install/initjvm.sql
CONNECT / AS SYSDBA
@/oracle/10202/db/xdk/admin/initxml.sql
CONNECT / AS SYSDBA
@/oracle/10202/db/rdbms/admin/catjava.sql
CONNECT / AS SYSDBA
@/oracle/10202/db/rdbms/admin/utlrp.sql
/*
```

In MetaLink Note 402242.1, the requirements of Data Pump are explained. If SQLORA13 was not run or failed, you can remedy the situation by following the MetaLink note. A summary of the note is given in Example 10-2.

Example 10-2 Summary of MetaLink Note 402242.1

```
   Subject: Ora-39213 Using Datapump Export On Z/Os
      Doc ID:
   Note:402242.1Type: PROBLEM
     Last Revision Date: 12-DEC-2006Status: MODERATED
    In this Document
     Symptoms
     Cause
     Solution

   This document is being delivered to you via Oracle Support's Rapid
   Visibility (RaV) Rapid Visibility (RaV) process, and therefore has
   not been subject to an independent technical review.
   Applies to:
   Operating System Dependent Interface - Version: 10.2.0.2.0
   This problem can occur on any platform.
   Symptoms
When running expdp the following errors are raised:

ORA-39006: internal error
```

```
ORA-39213: Metadata processing is not available
```

Cause

```
Datapump export uses DBMS_METADAT_UTIL package but this package does
not run succesfully.
 Executing sys.dbms_metadata_util.load_stylesheets procedure fails with
following error:
   ORA-31609: error loading file "kualter.xsl" from file system
directory "/rdbms/xml/xsl"
   ORA-06512: at "SYS.DBMS_METADATA_UTIL", line 1807
   ORA-06512: at line 1
This package needs XML being installed and XML needs JAVA being
installed.
Select * from dba_registry shows neither Java nor XML were installed:
  SQL> SELECT comp_id, version, status FROM dba_registry;
COMP_ID                        VERSION                        STATUS
-------------------------- -----------
    CATALOG                        10.2.0.2.0                     VALID
    CATPROC                        10.2.0.2.0                     VALID
```

Solution

```
1- Install JVM and XML  executing the following scripts from sqlplus:
CONNECT / as SYSDBA
/support/EMEA/10g/V10202/javavm/install/initjvm.sql
CONNECT / AS SYSDBA
/support/EMEA/10g/V10202/xdk/admin/initxml.sql
CONNECT / AS SYSDBA
CONNECT / AS SYSDBA
2- Check all scripts ran succesfully issuing select
comp_name,version,status from dba_registry :
   COMP_NAME                           VERSION
STATUS
   -------------
Oracle Database Catalog Views       10.2.0.2.0                     VALID
Oracle Database Packages and Types  10.2.0.2.0                     VALID
JServer JAVA Virtual Machine        10.2.0.2.0                     VALID
Oracle XDK                          10.2.0.2.0                     VALID
Oracle Database Java Packages       10.2.0.2.0                     VALID

3-Execute again sys.dbms_metadata_util.load_stylesheets from sqlplus :
CONNECT / AS SYSDBA
execute sys.dbms_metadata_util.load_stylesheets
```

10.5 Populating the database

SQL was used to create the tables used by the Web Application. The Web Application we used required several tables to be available. Example 10-3 shows the SQL used to create these tables.

Example 10-3 Create statements

```
CREATE TABLE HOLDINGEJB
  (PURCHASEPRICE DECIMAL(14, 2) NULL,
   HOLDINGID INTEGER NOT NULL,
   QUANTITY NUMBER NOT NULL,
   PURCHASEDATE DATE NULL,
   ACCOUNT_ACCOUNTID INTEGER NULL,
   QUOTE_SYMBOL VARCHAR2(250) NULL);

ALTER TABLE HOLDINGEJB
  ADD CONSTRAINT PK_HOLDINGEJB PRIMARY KEY (HOLDINGID);

CREATE TABLE ACCOUNTPROFILEEJB
  (ADDRESS VARCHAR2(250) NULL,
   PASSWD VARCHAR2(250) NULL,
   USERID VARCHAR2(250) NOT NULL,
   EMAIL VARCHAR2(250) NULL,
   CREDITCARD VARCHAR2(250) NULL,
   FULLNAME VARCHAR2(250) NULL);

ALTER TABLE ACCOUNTPROFILEEJB
  ADD CONSTRAINT PK_ACCOUNTPROFILEEJB PRIMARY KEY (USERID);

CREATE TABLE QUOTEEJB
  (LOW DECIMAL(14, 2) NULL,
   OPEN1 DECIMAL(14, 2) NULL,
   VOLUME NUMBER NOT NULL,
   PRICE DECIMAL(14, 2) NULL,
   HIGH DECIMAL(14, 2) NULL,
   COMPANYNAME VARCHAR2(250) NULL,
   SYMBOL VARCHAR2(250) NOT NULL,
   CHANGE1 NUMBER NOT NULL);

ALTER TABLE QUOTEEJB
  ADD CONSTRAINT PK_QUOTEEJB PRIMARY KEY (SYMBOL);

CREATE TABLE KEYGENEJB
  (KEYVAL INTEGER NOT NULL,
```

```
   KEYNAME VARCHAR2(250) NOT NULL);

ALTER TABLE KEYGENEJB
  ADD CONSTRAINT PK_KEYGENEJB PRIMARY KEY (KEYNAME);

CREATE TABLE ACCOUNTEJB
  (CREATIONDATE DATE NULL,
   OPENBALANCE DECIMAL(14, 2) NULL,
   LOGOUTCOUNT INTEGER NOT NULL,
   BALANCE DECIMAL(14, 2) NULL,
   ACCOUNTID INTEGER NOT NULL,
   LASTLOGIN DATE NULL,
   LOGINCOUNT INTEGER NOT NULL,
   PROFILE_USERID VARCHAR2(250) NULL);

ALTER TABLE ACCOUNTEJB
  ADD CONSTRAINT PK_ACCOUNTEJB PRIMARY KEY (ACCOUNTID);

CREATE TABLE ORDEREJB
  (ORDERFEE DECIMAL(14, 2) NULL,
   COMPLETIONDATE DATE NULL,
   ORDERTYPE VARCHAR2(250) NULL,
   ORDERSTATUS VARCHAR2(250) NULL,
   PRICE DECIMAL(14, 2) NULL,
   QUANTITY NUMBER NOT NULL,
   OPENDATE DATE NULL,
   ORDERID INTEGER NOT NULL,
   ACCOUNT_ACCOUNTID INTEGER NULL,
   QUOTE_SYMBOL VARCHAR2(250) NULL,
   HOLDING_HOLDINGID INTEGER NULL);

ALTER TABLE ORDEREJB
  ADD CONSTRAINT PK_ORDEREJB PRIMARY KEY (ORDERID);

CREATE INDEX a_profile_userid on accountejb(profile_userid);
CREATE INDEX h_account_accountid on holdingejb(account_accountid);
CREATE INDEX o_account_accountid on orderejb(account_accountid);
CREATE INDEX o_holding_holdingid on orderejb(holding_holdingid);
CREATE INDEX o_closed_orders on
orderejb(orderstatus,account_accountid);
```

Once we had the database, user, and tables created, we proceeded to populate the database and run the Web Application installation. After verifying that everything worked, we proceeded with the preparation to move the database.

10.6 Preparing for Data Pump

The next process was to use Data Pump to export the database from Oracle on z/OS and then import the database using Data Pump to an Oracle instance running on Linux for System z.

10.6.1 Data Pump Export and log files

The file that contains the database data written by Data Pump Export and read by Data Pump Import is called an *export file* or *dump file*. While conceptually similar to the export files of the Export and Import utilities, the two types are not interchangeable. For example, you cannot read a plain Export file with Data Pump Import and you cannot read a Data Pump Export file with plain Import.

Besides writing or reading an export file, the execution of Data Pump Export or Import normally writes a log file. This is just a text log reporting processing details.

On z/OS, the export and log files used by Data Pump Export and Import must be HFS file systems; they cannot be z/OS data sets. Because Data Pump export and log file processing is done in the Oracle database server rather than the client job or session, access to both types of file is controlled with database directory objects, which are created by a database administrator. Although there are forms of a directory object for both HFS files and data sets, only the HFS form of a directory object can be used in a Data Pump operation.

When we tried to do a complete database export with Data Pump, we had a failure on the import side. The solution was to include the parameter compression=none. This is explained in MetaLink Note 457255.1, as shown in Example 10-4.

Example 10-4 Data Pump MetaLink Note

```
Subject: DATAPUMP IMPORT FAILS WITH ORA-00600 [kupfiReadBzLob].
   Doc ID:
Note:457255.1Type: PROBLEM
   Last Revision Date: 05-SEP-2007Status: MODERATED
 In this Document
  Symptoms
  Cause
  Solution

This document is being delivered to you via Oracle Support's Rapid
Visibility (RaV) Rapid Visibility (RaV) process, and therefore has not
been subject to an independent technical review.
```

Applies to:
Oracle Server - Enterprise Edition - Version: 10.2
This problem can occur on any platform.

Symptoms
Datapump import is failing with the following set of errors :
Import: Release 10.2.0.3.0 - 64bit Production on Monday, 03 September, 2007 12:49:00

Copyright (c) 2003, 2005, Oracle. All rights reserved.
;;;
Connected to: Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options
Master table "STRMADMIN"."SYS_IMPORT_FULL_02" successfully loaded/unloaded
Starting "STRMADMIN"."SYS_IMPORT_FULL_02": strmadmin/******** FULL=y DIRECTORY=dpump_dir
DUMPFILE=tengd%U.dmp STREAMS_CONFIGURATION=n TRANSFORM=OID:N
Processing object type DATABASE_EXPORT/TABLESPACE
ORA-39014: One or more workers have prematurely exited.
ORA-39029: worker 1 with process name "DW01" prematurely terminated
ORA-31671: Worker process DW01 had an unhandled exception.
ORA-00600: internal error code, arguments: [kupfiReadBzLob], [kgcddo error], [19], [1], [2], [0],
[], []
ORA-06512: at "SYS.KUPW$WORKER", line 1342
ORA-06512: at line 2
Job "STRMADMIN"."SYS_IMPORT_FULL_02" stopped due to fatal error at 12:49:34

Cause

The default during the export is COMPRESSION=METADATA_ONLY which results in a compressed dump file. This compression is corrupted and causes the ORA-00600 and thus the import to fail.

Specifying COMPRESSION=NONE while performing the export should result in a successful import.

Solution

One workaround or could be a solution is to do the following :

1. Specifying COMPRESSION=NONE while performing the export.

```
As the default is COMPRESSION=METADATA_ONLY resulting in a compressed
dump file.
2. Importing the newly created non-compressed data file into the
destination DB.
```

10.6.2 Preparing to do the Data Pump export

The first thing to do is create a directory in z/OS UNIX that can be used for the exports and imports. After logging in as user ID oracle, go to a writable directory and create a directory:

```
mkdir expdata
```

We used SQL Plus to log into the database with the system user and create the directory that was used for the dump file and logs:

```
create directory mydata as '/u/oracle2/expdata';
```

10.7 Running the Data Pump export

We logged on to z/OS UNIX with the user ID oracle2 and ran the Data Pump export command to export the full database, as shown in Example 10-5.

Example 10-5 Running the Data Pump export

```
ORACLE2 @ SC53:/u/oracle2>expdp trade/trade
dumpfile=mydata:tradeschemanoc.dmp logfile=mydata:tradeschema.log
compression=none

Export: Release 10.2.0.2.0 - Production on Monday, 05 November, 2007
12:19:58

Copyright (c) 2003, 2005, Oracle.  All rights reserved.

Connected to: Oracle Database 10g Enterprise Edition Release 10.2.0.2.0
- Production
With the Partitioning, Oracle Label Security and Data Mining options
Starting "TRADE"."SYS_EXPORT_SCHEMA_01":  trade/********
dumpfile=mydata:tradeschemanoc.dmp logfile=mydata:tradeschema.log
compression=none
Estimate in progress using BLOCKS method...
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 640 KB
Processing object type SCHEMA_EXPORT/USER
```

```
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/TABLE/COMMENT
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
. . exported "TRADE"."ORDEREJB"                          152.6 KB
2462 rows
. . exported "TRADE"."HOLDINGEJB"                        86.25 KB
2462 rows
. . exported "TRADE"."QUOTEEJB"                          60.97 KB
1000 rows
. . exported "TRADE"."ACCOUNTEJB"                        29.89 KB
500 rows
. . exported "TRADE"."ACCOUNTPROFILEEJB"                 43.86 KB
500 rows
. . exported "TRADE"."KEYGENEJB"                        5.289 KB
3 rows
Master table "TRADE"."SYS_EXPORT_SCHEMA_01" successfully
loaded/unloaded
******************************************************************************
*******
Dump file set for TRADE.SYS_EXPORT_SCHEMA_01 is:
  /u/oracle2/tradeschemanoc.dmp
Job "TRADE"."SYS_EXPORT_SCHEMA_01" successfully completed at 12:20:25
```

Once the Data Pump Export finished, we did an **ls** command to check the files created; the output is shown in Example 10-6.

Example 10-6 Checking with the ls command

```
-rw-rw----   1 HAIMO    SYS1      524288 Oct 31 16:45 tradeschema.dmp
-rw-rw-r--   1 HAIMO    SYS1        1981 Nov  5 12:20 tradeschema.log
```

10.8 Moving the dump file to the Linux System

At this point, we needed to get the files to the new Linux Database Server. We used FTP, making sure we transmitted the files in binary format.

10.9 Installing Oracle 10g on Linux for System z

In order to use Data Pump, we need to have Oracle installed on our new Database server. We followed the instructions in Chapter 3, “Installing an Oracle 10gR2 single instance” on page 19.

10.10 Creating the database on Linux for System z

After the installation of the Oracle 10g binaries, we proceeded to create a database using the **dbca** command. This database was used to import the dump file created on Oracle on Z/OS. Make sure you create the database with the same specifications you had on Oracle on z/OS.

This chapter is not meant to cover the detailed installation process for a Oracle 10.2.0.2 database; that process is described in Chapter 3, “Installing an Oracle 10gR2 single instance” on page 19.

10.11 Preparing to do the Data Pump Import

On Linux for System z, we started with the Oracle installation and proceeded to create a database using the **dbca** command. Once we created the database, we continued to import, with Data Pump, the database we exported from the Oracle on z/OS system. It was also necessary to create the directory where we would place the dump file and log and create the object in the database:

```
mkdir expdata
```

We used SQL Plus to log into the database with the system user and create the directory that was used for the dump file and logs:

```
create directory mydata as '/home/oracle/expdata';
```

10.12 Running the Data Pump Import

The ID we used for the Data Pump Import was “system”, since the user and tables were not there yet. At the end, we logged on with the user “trade” to verify its creation, as shown in Example 10-7 on page 199.

Example 10-7 Running the Data Pump Import

```
oracle@linux24 expdata]$ impdp  system/manager
dumpfile=mydata:tradeschemanoc.dmp logfile=mydata:tradeschema1.log

Import: Release 10.2.0.2.0 - 64bit Production on Monday, 05 November,
2007 12:30:47

Copyright (c) 2003, 2005, Oracle.  All rights reserved.
Connected to: Oracle Database 10g Enterprise Edition Release 10.2.0.2.0
- 64bit Production
With the Partitioning, OLAP and Data Mining options
Master table "SYSTEM"."SYS_IMPORT_FULL_01" successfully loaded/unloaded
Starting "SYSTEM"."SYS_IMPORT_FULL_01":  system/********
dumpfile=mydata:tradeschemanoc.dmp logfile=mydata:tradeschema1.log
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
[oracle@linux24 expdata]$ impdp  system/manager
dumpfile=mydata:tradeschemanoc.dmp logfile=mydata:tradeschema1.log

Import: Release 10.2.0.2.0 - 64bit Production on Monday, 05 November,
2007 12:30:47
Copyright (c) 2003, 2005, Oracle.  All rights reserved.
Connected to: Oracle Database 10g Enterprise Edition Release 10.2.0.2.0
- 64bit Production
With the Partitioning, OLAP and Data Mining options
Master table "SYSTEM"."SYS_IMPORT_FULL_01" successfully loaded/unloaded
Starting "SYSTEM"."SYS_IMPORT_FULL_01":  system/********
dumpfile=mydata:tradeschemanoc.dmp logfile=mydata:tradeschema1.log
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. imported "TRADE"."ORDEREJB"                           152.6 KB    2462
rows
. . imported "TRADE"."HOLDINGEJB"                         86.25 KB
2462 rows
. . imported "TRADE"."QUOTEEJB"                           60.97 KB
1000 rows
. . imported "TRADE"."ACCOUNTEJB"                         29.89 KB
500 rows
```

```
. . imported "TRADE"."ACCOUNTPROFILEEJB"                 43.86 KB
500 rows
. . imported "TRADE"."KEYGENEJB"                         5.289 KB
3 rows
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Job "SYSTEM"."SYS_IMPORT_FULL_01" successfully completed at 12:31:16
```

10.13 Validating that the Data Pump was successful

Once the Data Pump completed successfully, we proceeded to verify that the database was imported successfully. The first thing we did was to start SQL PLUS and log on with the user ID and password that we used when logging into Oracle on z/OS. After we logged on, we proceeded to run the application and made sure that all functionality was there, as shown in Example 10-8.

Example 10-8 Verifying that the Data Pump was successful

```
[oracle@linux24 expdata]$ sqlplus

SQL*Plus: Release 10.2.0.2.0 - Production on Mon Nov 5 12:31:21 2007

Copyright (c) 1982, 2005, Oracle.  All Rights Reserved.

Enter user-name: trade
Enter password:

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - 64bit
Production
With the Partitioning, OLAP and Data Mining options

SQL> exit
Disconnected from Oracle Database 10g Enterprise Edition Release
10.2.0.2.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options
```

11

Using z/VM 5.3 LDAP Authentication for Oracle users

Chapter 9, “Using Radius Server and z/OS RACF LDAP for Oracle DB user authentication”, in *Experiences with Oracle 10g Database for Linux on zSeries*, SG24-6482 contains a discussion on the authentication of Oracle clients using an LDAP server running on z/OS. This chapter parallels that discussion. The major difference is that the LDAP server is running on z/VM V5.3 instead of z/OS.

The Oracle Client user ID and password authentication method discussed in this chapter is not restricted to a particular configuration. However, the method may provide more utility in a z/VM environment running multiple Linux systems, where some of the Linux systems are running Oracle Database servers. The usefulness of this method may be enhanced further, as the underlying z/VM 5.3 Lightweight Directory Access Protocol (LDAP) support could be extended to Linux and CMS users. For example, the z/VM LDAP server could be used to provide a single user name and password for an Oracle client, a Linux user, and a CMS user.

11.1 Introduction

An Oracle 10g database server was configured to do external password authentication of Oracle client user names and passwords. The Oracle Advanced Security Option (ASO) is required to be installed on both the Oracle client and database to do external password authentication. The sqlnet.ora files in both the database and client were configured to use a RADIUS server for the external authentication. The RADIUS server in turn was configured to use the z/VM LDAP server.

Example 11-1 shows an overview of the configuration.

Example 11-1 Overview of the configuration

```
Oracle Client <-->   Oracle DB    <--> Radius Server     <--> LDAP Server
Win/XP                 Linux25           Linux20             z/VM
```

The Oracle database is a RADIUS client and the RADIUS server is a client to the z/VM 5.3 LDAP Server. The details of configuring z/VM LDAP, RADIUS, and Oracle for Oracle external authentication are described in the following sections.

11.2 z/VM LDAP

The LDAP data model is closely aligned with the X.500 data model. In this model, a directory service provides a hierarchically organized set of entries. Each of these entries is represented by an object class and the object classes are part of a schema and the schema is available to the LDAP server. The object class of the entry determines the set of attributes that are required to be present in the entry as well as the set of attributes that can optionally appear in the entry. An attribute is represented by an attribute type and one or more attribute values. In addition to the attribute type and values, each attribute has an associated syntax that describes the format of the attribute values. Every entry in the directory has a distinguished name (DN). The DN is the name that uniquely identifies an entry in the directory. A DN is made up of attribute=value pairs, separated by commas. For example:

```
cn=iccuser1, ou=icc, o=ibm
```

is an LDAP DN. In addition to these lines:

```
DN: cn=iccuser2, ou=icc, o=ibm and DN: racfid=iccracf1,
profiletype=user, cn=racfvm, o=ibm, c=us
```

it will be used in the following discussion.

It is assumed that there is a LDAP server running on a z/VM V5.3 system and that the server can access both LDBM and SDBM back ends to store and retrieve LDAP directory information. The GDBM back end, which can be used to implement a change log, is not discussed. The references given below cover setting up a z/VM LDAP server, so the discussion here is about the specific configuration that needs to be done so that a communication path is available for user authentication between Oracle and the z/VM LDAP server.

11.2.1 z/VM LDBM

z/VM LDAP is a server function of TCP/IP provided in z/VM V5.3 and it runs as a virtual machine under the OPENVM SHELL. The z/VM Lightweight Directory Access Protocol (LDAP) server is based on a client-server model that provides client access to an LDAP server. The server provides a way to maintain directory information in a central location. z/VM supports two back ends or databases that can contain LDAP directory information. The back end discussed in this section is the LDBM, which is a general purpose back end. It uses the z/VM CMS Byte File System (BFS) to store directory information. In general, when the z/VM LDAP server is running the LDBM directory is maintained in storage.

Even though it is assumed here that the z/VM LDAP server is up and running, three items related to the server are described briefly. By default, the z/VM LDAP server configuration parameters are in a CMS file, DS CONF, which is located on the TCPMAINT 198 minidisk. The default can be changed by a command-line parameter when the server's virtual machine is started, but the default is used here. References to several parameters in the DS CONF file and their specific settings will be made. Secondly, from CMS, the information in the LDBM directory is accessed and managed by LDAP utilities (commands) such as **ldapadd** and **ldapsearch**. These utilities are packaged as EXECs in CMS and the EXECs are on the TCPMAINT 592 disk.

The utilities follow the same syntax as would be used if running them in other systems, except that the names invoked are CMS EXECs, so the invocation names are eight characters. Thus, the EXEC name for the ldapsearch utility is LDAPSRCH. Also, these utilities, whether from CMS or Linux, are clients to the z/VM LDAP server and the client/server connection can be secured with SSL/TLS, but SSL/TLS is not used here. Finally, access to LDAP directory entries and attributes is controlled by the z/VM LDAP server administrator and Access Control Lists (ACLs). The ACLs are attributes in the directory entries and are managed by the LDAP utilities mentioned previously. For the discussion that follows, the ACL defaults are used, which means that all users can read, search, and compare the LDAP entries in the LDBM, but only the DN defined in the adminDN parameter in the DS CONF file and the DN that is propagated down the LDAP hierarchy to the entries entryOwner attribute can change the entry.

The Oracle user name iccuser1 was added to the LDBM directory with the CMS EXEC:

```
ldapadd  -h 9.12.4.92 -D "cn=ldapadm4,o=ibm"
-w ldapadmx -f //iccuser.ldif
```

The content of the ICCUSER LDIF A CMS file was:

```
dn: cn=iccuser1, ou=icc, o=ibm
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: posixAccount
cn: iccuser1
sn: iccuser1
uid: iccuser1
userPassword: iccuser1x
uidNumber: 50001
gidNumber: 500
homedirectory: /home/iccuser1
loginshell: /bin/bash
```

The adminDN entry in the LDAP server DS CONF file is cn=ldapadm4,o=ibm. There is no particular reason that the administrator DN has only two levels in the directory and the iccuser1 DN being added has three levels other than it seemed like a logical organizational structure. The directory levels were also created with the CMS LDAPADD EXEC.

Retrieving the entry from the directory with the CMS command:

```
ldapsrch -h 9.12.4.92  -D "cn=ldapadm4,o=ibm"
   -w ldapadmx -b "cn=iccuser1,ou=icc,o=ibm"
   "objectclass=*"
```

yields the following output as expected:

```
cn=iccuser1, ou=icc, o=ibm
objectclass=top
objectclass=person
objectclass=organizationalPerson
objectclass=posixAccount
cn=iccuser1
sn=iccuser1
uid=iccuser1
userpassword=iccuser1x
uidnumber=50001
gidnumber=500
homedirectory=/home/iccuser1
```

```
loginshell=/bin/bash
```

The userPassword is entered as clear text and will remain that way in the directory unless the password encryption option, pwEncryption, in the DS CONF file is set to encrypt the userPassword attribute. If the pwEncryption parameter setting is changed from none to SHA, the userPassword is encrypted by the SHA algorithm. For entries that have already been entered into the directory without encryption, there is a utility, **db2pwden**, that can be used to encrypt existing clear text passwords. A simple **ldapsrch** search of an entry with an encrypted userPassword results in userpassword=NOT Printable being displayed. What actually is returned is a tag indicating SHA and the encrypted password, which is not printable to CMS. This encryption applies only to the LDBM database.

The options -D cn=ldapadm4,o=ibm and -w ldapadmx on the CMS LDAPSRCH above are required, as the DS CONF file parameter allowAnonymousBinds is set to off, so anonymous binds are not allowed. The ldapadm4 name with password ldapadmx is used to explicitly bind to the LDAP server. If the allowAnonymousBinds parameter is set to on and anonymous binds are allowed, then the LDAPSRCH will work without the -D and -w options being specified.

The LDBM back end also used the following settings in the z/VM LDAP server DS CONF file:

```
database LDBM GLDBLD31
suffix "o=ibm"
```

11.2.2 z/VM LDBM and native authentication

The z/VM LDAP server allows clients to bind to entries in an LDBM back end while using RACF for verifying the supplied password. The entry for the Oracle user name, except for the password, is maintained in LDBM, just as specified in the LDBM case for iccuser1. There are a few differences in both the LDBM directory entry and the z/VM LDAP server configuration file. iccuser2 is the Oracle user name used for this method of authentication using z/VM LDAP. The directory entry for iccuser2 is shown below.

```
cn=iccuser2, ou=icc, o=ibm
objectclass=top
objectclass=person
objectclass=organizationalPerson
objectclass=ibm-nativeAuthentication
objectclass=posixAccount
cn=iccuser2
sn=iccuser2
ibm-nativeid=ICCUSER2
uid=iccuser2
```

```
uidnumber=50002
gidnumber=500
homedirectory=/home/iccuser2
loginshell=/bin/bash
```

Notice that there is no password attribute and value pair in the LDBM directory entry. The objectclass=ibm-nativeAuthentication entry is specified and its definition includes the ibm-nativeid attribute. The objectclass allows the use of the attribute ibm-nativeid and this is the connection to RACF for z/VM LDAP native authentication. The RACF ID must be added in uppercase to the directory entry; otherwise, the native authentication will fail.

iccuser2 was added to z/VM RACF with the following RACF commands from CMS:

```
rac au (iccuser2) pass(password)
rac alu (iccuser2) noexpired password(ldap22)
```

The **alteruser** command was run to prevent having to change the password upon the first access to RACF by iccuser2.

Native authentication used the following three parameter settings in the z/VM LDAP DS CONF file:

```
useNativeAuth selected
nativeUpdateAllowed on
nativeAuthSubtree "ou=icc,o=ibm"
```

11.2.3 z/VM SDBM

The z/VM LDAP server provides access to RACF user information stored in the RACF SDBM database. RACF group and connection information is also available, but is not discussed in this document. SDBM is also known as the RACF database back end of the LDAP server. The SDBM database allows for directory authentication using a RACF user ID and password. z/VM RACF must be available to allow the LDAP server to communicate with RACF and the LDAP server must be configured to use SDBM. The format of the DN is restricted when using SDBM, as the DN must match the schema of the underlying RACF data. A RACF SDBM style DN for a user contains two required attributes plus a suffix. A valid DN for RACF is racfid=userid,profiletype=user,suffix. The required attributes are racfid and profiletype. For the discussion here, the attribute value of the profiletype attribute will always be user. In other contexts, group or connect are valid for the profiletype attribute.

The DN suffix for SDBM used here is cn=racfvm, o=ibm, c=us. A complete DN for the Oracle user in the examples here is DN: racfid=iccracf1, profiletype=user,

cn=racfvm, o=ibm, c=us. The z/VM LDAP server can be configured to support LDBM and SDBM at the same time, but for the discussion here only the SDBM is configured in the DS CONF file and the administrator is also configured as a RACF DN. The following are the DS CONF file settings for the RACF SDBM setup:

```
adminDN  "racfid=ldapadm2,profileType=user,cn=racfvm,o=ibm,c=us"
database SDBM GLDBSD31
suffix "cn=racfvm,o=ibm,c=us"
```

The z/VM LDAP server administrator, ldapadm2, was added to RACF with the **adduser** command and the directory structure was created from the suffix when the LDAP server was first started with the above values in the DS CONF file.

The Oracle user name, iccracf1, was added to the z/VM LDAP RACF SDBM directory with the following command:

```
ldapadd  -h 9.12.4.92  -D "racfid=ldapadm2, profiletype=user,
cn=racfvm, o=ibm, c=us"  -w ldap00 -f //iccracf.ldif
```

The contents of the CMS ICCRACF LDIF file are:

```
dn: racfid=iccracf1,profiletype=user,cn=racfvm,o=ibm,c=us
objectclass: racfUser
racfid: iccracf1
```

The RACF **alteruser** CMS command **rac alu (iccracf1) noexpired password(ldapxx)** was run to add a password to iccracf1. Now everything about the Oracle user name, iccracf1, exists in the RACF SDBM database.

The CMS EXEC LDAPSRCH:

```
ldapsrch -h 9.12.4.92 -D "racfid=ldapadm2, profileType=User, cn=racfvm,
o=ibm, c=us"  -w ldap00 -s sub  -b "racfid=iccracf1, profileType=user,
cn=racfvm, o=ibm, c=us" "objectclass=*"
```

returns the following for iccracf1 user:

```
racfid=ICCRACF1,profiletype=USER,cn=racfvm,o=ibm,c=us
racfid=ICCRACF1
racfauthorizationdate=12/18/07
racfowner=RACFID=LDAPADM2,PROFILETYPE=USER,CN=RACFVM,O=IBM,C=US
racfpasswordinterval=30
racfpasswordchangedate=12/18/07
racfdefaultgroup=RACFID=SYS1,PROFILETYPE=GROUP,CN=RACFVM,O=IBM,C=US
racflastaccess=01/08/08/12:52:57
racflogondays=SUNDAY
racflogondays=MONDAY
racflogondays=TUESDAY
```

```
racflogondays=WEDNESDAY
racflogondays=THURSDAY
racflogondays=FRIDAY
racflogondays=SATURDAY
racflogontime=ANYTIME
racfconnectgroupname=RACFID=SYS1,PROFILETYPE=GROUP,CN=RACFVM,O=IBM,C=US
racfhavepasswordenvelope=NO
racfattributes=PASSWORD
objectclass=RACFBASECOMMON
objectclass=RACFUSER
```

11.2.4 References for LDAP on z/VM V5.3

These documents were used to implement the tasks in this section:

- *Security on z/VM*, SG24-7471
- *z/VM TCP/IP LDAP Administration Guide*, SC24-6140
- *z/VM TCP/IP Planning and Customization Guide,* SC24-6125
- *z/VM TCP/IP User's Guide,* SC24-6127
- *z/VM RACF Security Server Command Language Reference,* SC24-6144

11.3 FreeRADIUS on Linux for System z

RADIUS is a client-server security protocol that can be used by the Oracle Advanced Security Option (ASO) to do external password authentication. There are several RADIUS servers available and they can be found by searching the Internet. The FreeRADIUS server available from `http://www.freeradius.org` was installed and configured on SLES9 Linux for System z. The SLES9 distribution CDs contain the freeradius-1.0.5-2.4 package, and this is the package that was used.

11.3.1 FreeRADIUS installation

A standard **`rpm -ivh`** install of the freeradius-1.0.5-2.4 package was done.

11.3.2 FreeRADIUS configuration

FreeRADIUS was configured to operate as a server for the Oracle Database and as a client to the z/VM LDAP server. Three FreeRADIUS configuration files, radiusd.conf, clients.conf, and users, were modified to support the configurations

discussed in this document. The following details show the specific pieces of each file that was changed from the FreeRADIUS default values. After the installation of FreeRADIUS, these file are all located in the /etc/raddb directory.

11.3.3 FreeRADIUS configuration radiusd.conf file

The radiusd.conf file was set up in three different ways to show three ways to use the z/VM LDAP server for user authentication from FreeRADIUS. In all cases, the z/VM LDAP server was identified by a network name and the default port of 389. The vmlinux9 name used here was at IP address 9.12.4.92. It was used in the discussion above and either can be used in either place.

The first two Oracle clients, iccuser1 and iccuser2, are in the z/VM LDAP under ou=icc, o=ibm in the Directory Information Tree (DIT). This hierarchy value was used as the FreeRADIUS *basedn* value in the radiusd.conf file. The basedn is where the search will start in the LDAP directory for the Oracle user names. The Oracle name, iccuser1, has a DN of cn=iccuser1, ou=icc, o=ibm and a password of iccuser1x. The details of setting up Oracle for external authentication will be given in later sections. Also, what is shown in Example 11-2 and Example 11-3 on page 210 is just a snippet of the FreeRADIUS radiusd.conf configuration file.

If the z/VM LDAP server allows anonymous binds with the parameter setting, allowAnonymousBinds on in the DS CONF file, then the following will allow FreeRADIUS to connect to the z/VM LDAP Server for password authentication of the Oracle client.

Example 11-2 Radius.conf file snippet part 1

```
ldap {
                server = "vmlinux9.itso.ibm.com"
                basedn = "ou=icc,o=ibm"
                password_attribute = "userPassword"
                filter ="(uid=%{Stripped-User-Name:-%{User-Name}})"
        }
Auth-Type LDAP {
                 ldap
               }
```

If z/VM LDAP does not allow anonymous binds and has the parameter setting “allowAnonymousBinds off”, then FreeRADIUS needs to have a user ID and password (ldapam4/ldapadmx) that it can send to the z/VM LDAP server. This allows it to do a bind to the z/VM LDAP server before the search for the password for authentication of the Oracle user is done. The bind user ID/password is provided in the configuration file with the identify and password settings. The

password is clear text, so appropriate permission settings need to be maintained in the configuration file.

Example 11-3 Radius.conf file snippet part 2

```
ldap {
                server = "vmlinux9.itso.ibm.com"
                identity = "cn=ldapadm4,o=ibm"
                password = ldapadmx
                basedn = "ou=icc,o=ibm"
                password_attribute = "userPassword"
                filter= “(uid=%{Stripped-User-Name:-%{User-Name}})"
       }
Auth-Type LDAP {
                ldap
                }
```

If the z/VM LDAP server is using the RACF SDBM database to contain Oracle user names and passwords, then a slightly different radiusd.conf file is needed, as shown in Example 11-4. For RACF users, the LDAP DN is fixed to a racfid, a profileType=user, and a suffix. It is also possible to maintain RACF group and connection information in the SDBM back end, but they are not used or discussed here. The DN for the Oracle ID, iccracf1, is DN: racfid=iccracf1, profileType=user, cn=racfvm, o=ibm, c=us. The suffix is cn=racfvm, o=ibm, c=us. ldapadm2 is a RACF user ID that can bind to the z/VM LDAP server and search RACF for the iccracf1 user name. As above, the password is clear text, so appropriate permission settings needs to be maintained on the configuration file.

Example 11-4 Radius.conf file snippet part 3

```
ldap {
        server = "vmlinux9.itso.ibm.com"
        identity =
          "racfid=ldapadm2,profiletype=user,cn=racfvm,o=ibm,c=us"
       password = ldap00
        basedn = "racfid=%{Stripped-User-Name:-%{User-
         Name}},profiletype=user,cn=racfvm,o=ibm,c=us"
        password_attribute = "userPassword"
        filter = "(objectclass=*)"
      }
Auth-Type LDAP {
                    ldap
               }
```

11.3.4 FreeRADIUS Configuration clients.conf file

An entry was added to the clients.conf file to identify the network name of the Oracle Database server that will connect to FreeRADIUS as a RADIUS client to do external authentication for Oracle user names. The default port is 1812. The Radius Server secret is specified in this file entry. This is the key that the Oracle Database and FreeRADIUS use to encrypt and decrypt the Oracle password that is passed between FreeRADIUS and Oracle. The secret is made available to Oracle when it is set up for external authentication and the detail is provided in a later section.

Note: There are various descriptions of the specification of the Radius secret and it took some experimenting before a key was found that would successfully work between the Oracle Database and FreeRADIUS. The encrypt/decrypt between Oracle and FreeRADIUS did not seem to work if the key is longer than 15 characters. The radtest program mentioned below worked fine with a key longer than 15 characters:

```
client   linux25.itso.ibm.com
{
         secret = testing-123
         shortname =  linux25
}
```

11.3.5 FreeRADIUS Configuration users file

The third file, *users*, identifies the Oracle users (iccuser1, iccuser2, and iccracf1) that will authenticate using the z/VM LDAP server. These user names will be used in later examples in this chapter. These are the same names that were used above in the setup of the z/VM LDAP server:

```
iccuser1 Auth-Type = LDAP
iccuser2 Auth-Type = LDAP
iccracf1 Auth-Type = LDAP
```

At this point, the FreeRADIUS server is started on linux20 by executing the binary radiusd. Now both FreeRADIUS and the z/VM LDAP server are configured, running, and awaiting user requests from Oracle for external authentication.

11.3.6 FreeRADIUS debug mode

FreeRADIUS can be started in debug mode by supplying the -X and -A options, for example, /usr/local/sbin/radiusd -X -A. This provides a great deal of information and is particularly helpful when the conf files above are changed, as much of the information from the files is printed when FreeRADIUS is started. In this mode, the information about the FreeRADIUS connections to both the Oracle DB and z/VM LDAP server is also printed.

11.3.7 FreeRADIUS test driver

FreeRADIUS provides a test driver, **radtest**, that is useful in testing that FreeRADIUS is installed and configured correctly, for example, executing **/usr/local/bin/radtest iccuser1 iccuser1x 127.0.0.1 0 testing-123** tests that iccuser1 is in the users file, tests the encrypt and decrypt of the password iccuser1x with the secret of testing-123, and tests that FreeRADIUS sends the user ID/password, iccuser1/iccuser1x, to z/VM LDAP for authentication. This example shows a local test, but if the binary is available, **radtest** can be run from a remote system. The IP address must be adjusted according to where the FreeRADIUS server is running.

11.3.8 References for FreeRADIUS

There is a lot of Radius Server information available on the Internet. But the README files, other referenced files, and comments in the configuration files (particularly radiusd.conf) are the best source of detailed documentation for FreeRADIUS.

11.4 Oracle setup for external authentication

RADIUS support is installed as a part of Oracle Advanced Security during the install of the Oracle Database. Remote Authentication Dial-In User Service (RADIUS) is a client-server security protocol that is used for remote authentication and access. Oracle Advanced Security uses this standard in a client-server network environment to enable use of any authentication method that supports the RADIUS protocol.

11.4.1 Oracle Database setup for FreeRADIUS

After the database is installed with the advanced security option, there are three steps that need to be done on the Oracle Database server so that an Oracle user can be externally authenticated through a RADIUS server. The first step is to create the RADIUS secret file for the database server. The contents of the file must match the value specified for the secret parameter in the FreeRADIUS clients.conf file that was set up and described in 11.3.4, "FreeRADIUS Configuration clients.conf file" on page 211:

```
# cat /oracle1/db1/network/radius.key
testing-123
#
```

The next step is to use Oracle Net Manager or directly edit the sqlnet.ora file so that the database server can use FreeRADIUS for external authentication. The FreeRADIUS server is running on network address linux20.itso.ibm.com and listening on port 1812. Also, sqlnet.ora contains the location of the secret file:.

```
#cat /oracle1/db1/network/admin/sqlnet.ora
# sqlnet.ora Network Configuration File:
#/oracle1/db1/network/admin/sqlnet.ora
# Generated by Oracle configuration tools.
SQLNET.AUTHENTICATION_SERVICES= (BEQ,TCPS,NTS,RADIUS)
SQLNET.RADIUS_SECRET = /oracle1/db1/network/radius.key
NAMES.DIRECTORY_PATH= (TNSNAMES,EZCONNECT)
SQLNET.RADIUS_AUTHENTICATION_TIMEOUT = 10
SQLNET.RADIUS_AUTHENTICATION = linux20.itso.ibm.com
SQLNET.RADIUS_AUTHENTICATION_PORT = 1812
SQLNET.RADIUS_SEND_ACCOUNTING = OFF
SQLNET.RADIUS_CHALLENGE_RESPONSE = OFF
#
```

The final step on the database server to use external authentication is to create the Oracle users that will use the FreeRADIUS server. When you choose external authentication for a user, the user account is maintained by Oracle, but password administration and user authentication is performed by an external service or FreeRADIUS in combination with the z/VM LDAP server here. The following statements create users who are identified by Oracle and authenticated by FreeRADIUS. The users created here are referenced throughout the document as examples.

The setup used here also has the following statement, os_authent_prefix = "", in the Oracle Database initialization file or pfile so that there is not any prefix:

```
SQL> create user iccuser1 identified externally;
User created.
SQL> grant create session to iccuser1;
Grant succeeded.
SQL> create user iccuser2 identified externally;
User created.
SQL> grant create session to iccuser2;
Grant succeeded.
SQL> create user iccracf1 identified externally;
User created.
SQL> grant create session to iccracf1;
Grant succeeded.
SQL>
```

11.4.2 Oracle client

The Oracle client installation on WIN/XP requires the selection and installation of the Advanced Security Option to be able to do RADIUS authentication from the Oracle client.

The Oracle Net Manager is used to configure the Oracle Client to use RADIUS for authentication. From the local profile window, shown in Figure 11-1 on page 215, select **Oracle Advanced Security** in the drop-down menu, select the **Authentication** tab, and move RADIUS from the Available Methods pane to the Selected Methods pane.

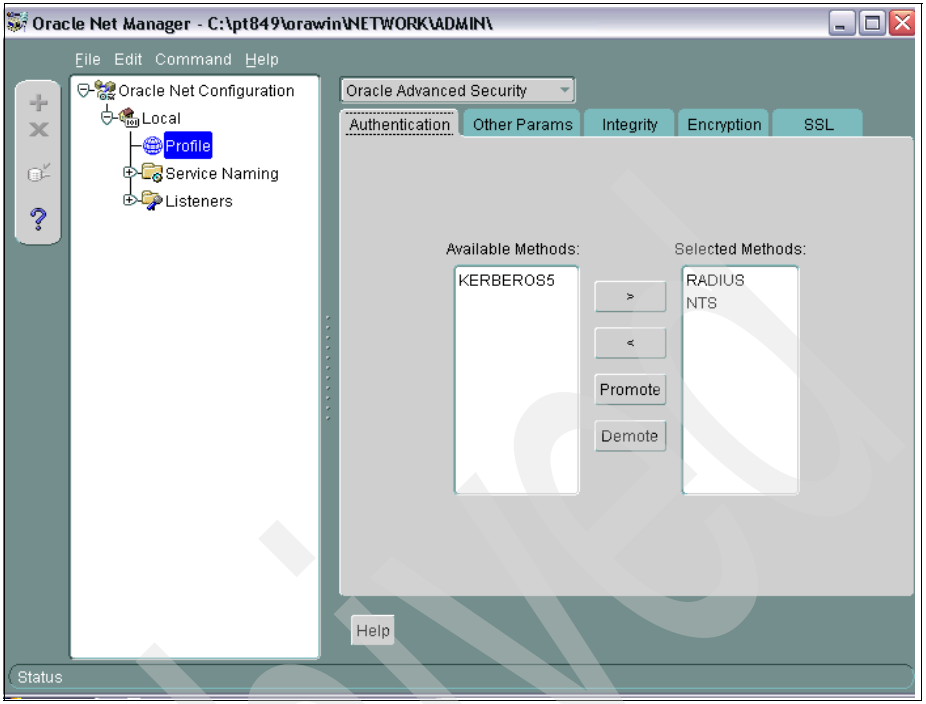


Figure 11-1 Oracle Advanced Security window

This will update the client sqlnet.ora file by adding RADIUS to the SQLNET.AUTHENTICATION_SERVICES parameter:

```
# sqlnet.ora Network Configuration File:
C:\pt849\orawin\NETWORK\ADMIN\sqlnet.ora
# Generated by Oracle configuration tools.
SQLNET.AUTHENTICATION_SERVICES= (RADIUS,NTS)
NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
```

11.4.3 Oracle Client Connection with User Name in LDBM and Password in SDBM

At this point, the Oracle user, iccuser2, can do a normal connection to the Oracle Database server and will be authenticated by z/VM LDAP as the back end to the FreeRADIUS server. The radiusd.conf file being used is either that shown in Example 11-2 on page 209 or Example 11-3 on page 210, depending on whether anonymous binds are allowed. The iccuser2 is shown in the windows here and it was set up for z/VM LDAP native authentication, meaning that its password is in RACF and not in the LDBM directory. The same windows are available for iccuser1 and its password is in the LDBM directory.

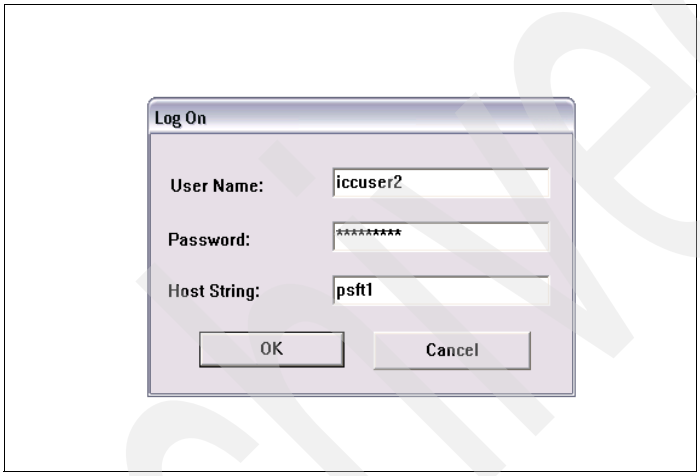


Figure 11-2 Logon window

After entering the user name, the valid password, and the Oracle connect string, the user is connected to the Oracle Database, as shown in Figure 11-3.

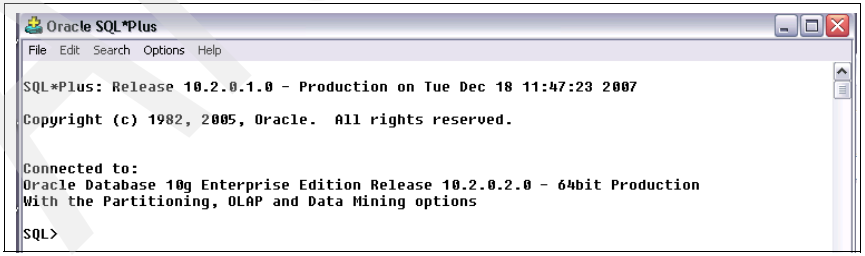


Figure 11-3 SQL connection

If the incorrect password is entered for iccuser2, the window shown in Figure 11-4 on page 217 appears.

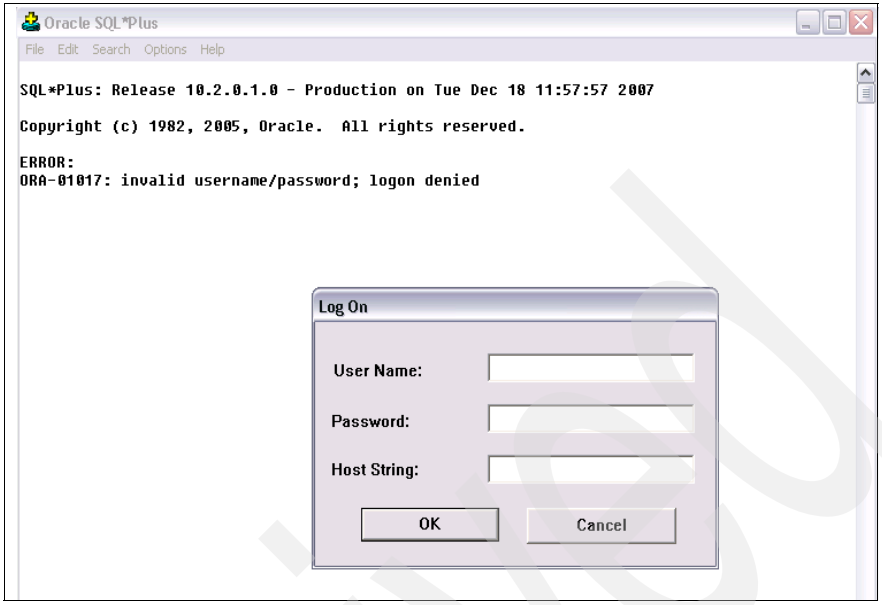


Figure 11-4 Incorrect password message

11.4.4 Oracle client connection with user name and password in SDBM

The Oracle user, iccracf1, and password, ldapxx, are in the RACF SDBM database. The only difference when the Oracle client is connecting to the database is iccracf1/ldapxx is used instead of iccuser1/iccuser1x. The major difference is that the FreeRADIUS server needs to be started with a radiusd.conf file, shown in Example 11-4 on page 210. The parameter settings in this file allow FreeRADIUS to bind to z/VM LDAP with the RACF SDBM back end and do the search for iccracf1 in RACF for user authentication.

11.4.5 Oracle references for external authentication

- *Oracle Database Security Guide 10g Release 2*, B14266-01, June 2005
- *Oracle Database Advanced Security Administrator's Guide*, B14268-02, November 2005

Abbreviations and acronyms

ADF	Application Development Framework
API	Application Programming Interface
AS	Application Server
ASM	Automated Storage Manager
BIEE	Business Intelligence Enterprise Edition
BPEL	Business Process Execution Language
CRS	Cluster Ready Services
CVU	Cluster Verification Utility
CW	ClusterWare
DB	Database
EBS	E-Business Suite
GSD	Global Services Daemon
HTTP	Hypertext Transport Protocol
IBM	International Business Machines Corporation
IM	Identity Manager
ITSO	International Technical Support Organization
J2EE	Java 2 Enterprise Edition
JDK	Java Developer Kit
JMS	Java Message Service
LDAP	Lightweight Directory Access Protocol
LVM	Logical Volume Manager
OCFS2	Oracle Cluster File System 2
OCR	Oracle Cluster Registry
OCS	Oracle Collaboration Suite
OEM	Oracle Enterprise Manager
OFA	Optimum Flexible Architecture
ONS	Oracle Notification Service
OTN	Oracle Technical Network
OUI	Oracle Universal Installer
RAC	Real Application Clusters
RPM	Red Hat Package Manager
SID	System Identifier
SOA	Service-Oriented Architecture
SQL	Structured Query Language
SR	Oracle Service Request
VIP	Virtual Cluster IP addresses
VM	Virtual Machine
VNC	Virtual Network Computing
XDK	XML Developer Kit
XML	EXstensible Markup Language

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 223. Note that some of the documents referenced here may be available in softcopy only.

- *Building Linux Systems Under IBM VM*, REDP-0120
- *Experiences with Oracle 10g Database for Linux on zSeries*, SG24-6482
- *Experiences with Oracle?® 10gR2 Solutions on Linux for IBM System z*, SG24-7191
- *Experiences with Oracle for Linux on zSeries*, SG24-6552
- *Linux Handbook A Guide to IBM Linux Solutions and Resources*, SG24-7000
- *Linux for IBM eServer zSeries and S/390: Distributions, SG24-6264*
- *Linux for S/390*, SG24-4987
- *Linux on IBM eServer zSeries and S/390: System Management*, SG24-6820
- *Linux on IBM eServer zSeries and S/390: ISP/ASP Solutions*, SG24-6299
- *z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES9*, SG24-6695

Other publications

These publications are also relevant as further information sources:

- *10gR2 Installation Guide for IBM zSeries Based Linux,* B25400-01
- *10gR2 Quick Installation Guide for IBM zSeries Based Linux,* B28935-01
- *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide 10g Release 2 (10.2),* B14197-03
- *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide 10g Release 2 (10.2) for Linux,* B14203-08

- *Oracle Database Release Notes 10g Release 2 (10.2) for IBM zSeries Based Linux*, B25399-04
- *MetaLink Notes*
  - *Note 220970.1 RAC Frequently Asked Questions Topics*
  - *Note 239998.1 10g RAC How to clean up after a failed CRS Install*
  - *Note 259301.1 CRS and 10g Real Application Clusters*
  - *Note 268937.1 Repairing or Restoring an Inconsistent OCR in RAC*
  - *Note 270577.1 Installing Oracle 9i on IBM z/Series - SLES8/9*
  - *Note 415182.1 DB Install Requirements Quick Reference - zSeries based Linux.*
  - *Note 417001.1 Errors installing 10.2.0.2 patchset on IBM ZSeries Based Linux.*
  - *Note 420382.1 Requirements for Installing Oracle 10gR2 RDBMS on RHEL 4 on zLinux (s390x).*
  - *Note 431443.1 Requirements for Installing Oracle 10gR2 RDBMS on SLES 9 zLinux (s390x)*

Online resources

These Web sites are also relevant as further information sources:

- Oracle Corporate URL

  `http://www.oracle.com`
- Oracle MetaLink - for Oracle support

  `http://metalink.oracle.com`
- Oracle RAC User Group

  `http://www.oraclesigrac.org`
- Oracle Technology Network

  `http://otn.oracle.com`

How to get IBM Redbooks publications

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

E

F

G

H

I

P

Q

R

S

T

U

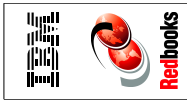
V

W

X

Y

Z



Using Oracle Solutions on Linux for System z

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Using Oracle Solutions on Linux for System z

Setting up VM and Linux for Oracle Database 10gR2

Setting up an environment for Data Guard

Setting up multiple Linux guests for Oracle

Linux on System z offers many advantages to customers who rely upon the IBM mainframe systems to run their businesses. Linux for System z takes advantage of the qualities of service in the System z hardware and in z/VM, making it a robust industrial strength Linux. This provides an excellent platform for hosting Oracle solutions that run in your enterprise.

This IBM Redbooks publication describes experiences gained while installing and testing several Oracle features and solutions, such as:

- Oracle Data Guard
- IBM WebSphere Application Server connecting to an Oracle database on Linux for System z
- Oracle BIEE connecting to an Oracle database on Linux for System z
- Oracle Data Pump

It also includes many general hints and tips on running Oracle products on IBM System z with Linux and z/VM.

The audience for this book includes database consultants, installers, administrators, and system programmers. This is not meant to replace Oracle documentation. It is just our experiences with installing and using Oracle products.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7573-00 ISBN 0738485985