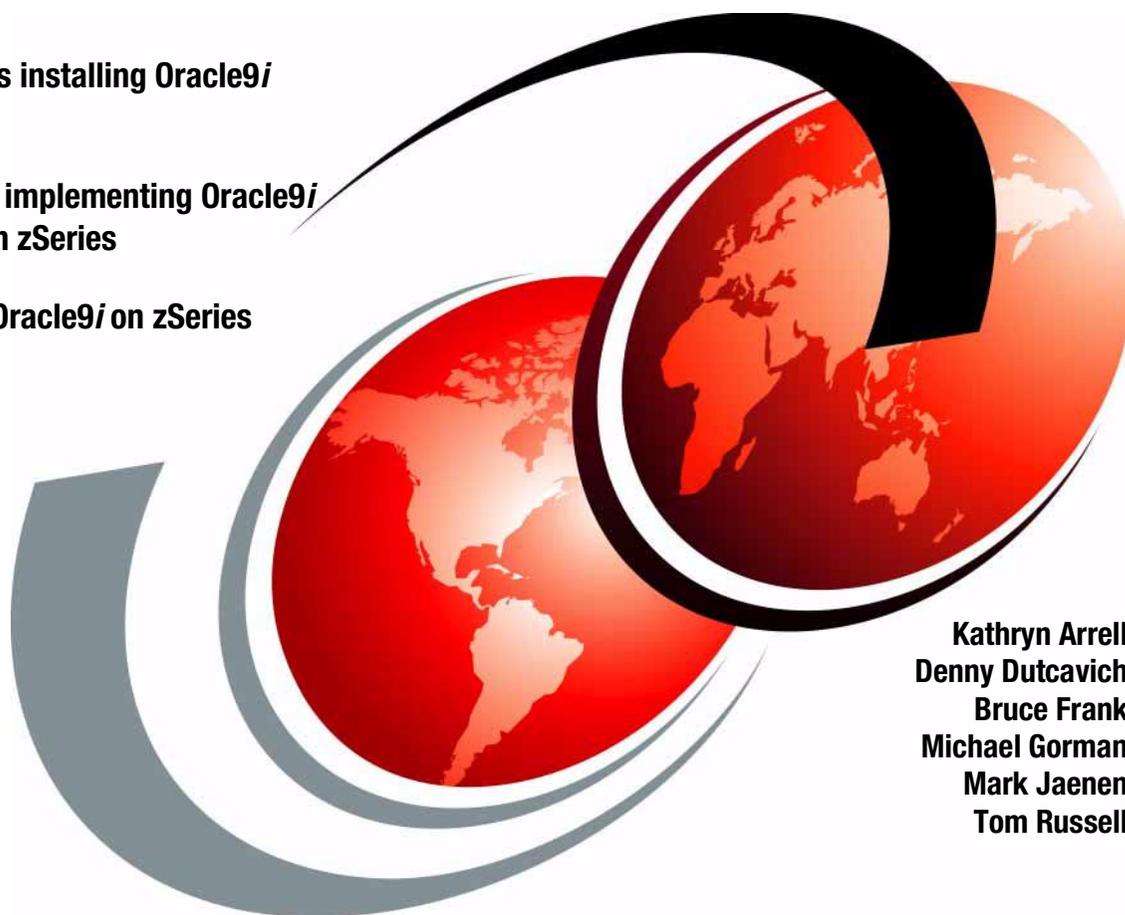


# Experiences with Oracle for Linux on zSeries

Experiences installing Oracle9i

Options for implementing Oracle9i  
for Linux on zSeries

Managing Oracle9i on zSeries



Kathryn Arrell  
Denny Dutcavich  
Bruce Frank  
Michael Gorman  
Mark Jaenen  
Tom Russell





International Technical Support Organization

**Experiences with Oracle for Linux on zSeries**

April 2003

**Take Note!** Before using this information and the product it supports, be sure to read the general information in “Notices” on page ix.

**First Edition (April 2003)**

This edition applies Oracle9i for Linux on zSeries for use with IBM zSeries hardware.

Comments may be addressed to:  
IBM Corporation, International Technical Support Organization  
Dept. HYJ Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2003. All rights reserved.

Note to U.S Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Notices</b> .....	ix
Trademarks .....	x
<b>Preface</b> .....	xi
The team that wrote this redbook .....	xi
Notice .....	xii
Comments welcome .....	xiii
<b>Chapter 1. Overview of Oracle9i on zSeries</b> .....	1
1.1 What is Linux .....	2
1.2 What is Linux for zSeries .....	2
1.3 Why Oracle9i on Linux for zSeries? .....	3
1.3.1 Expanded application portfolio .....	4
1.3.2 Cost savings .....	4
1.3.3 Server consolidation and faster time to market .....	5
1.4 What is Oracle9i for Linux on zSeries .....	7
1.4.1 Oracle9i Database Server .....	7
1.4.2 Oracle9i Application Server .....	8
1.4.3 The Oracle e-business Suite .....	8
1.4.4 The Oracle Collaboration Suite .....	8
1.5 What distributions of Linux does Oracle support? .....	8
1.6 Obtaining a copy of Oracle9i for Linux on zSeries .....	8
1.7 Sizing Workloads for Oracle9i on Linux for zSeries .....	9
<b>Chapter 2. How to deploy Oracle9i on Linux/390</b> .....	11
2.1 Deployment options .....	12
2.1.1 z/VM or LPAR? .....	12
2.1.2 CPs or IFLs? .....	15
2.1.3 SuSE or Red Hat? .....	15
2.1.4 Network topology .....	16
2.1.5 Capacity planning .....	20
2.1.6 Customization of the SuSE distribution .....	25
<b>Chapter 3. Installing Oracle9i Database on zSeries with Linux</b> .....	27
3.1 The system environment .....	28
3.2 The Linux environment .....	28
3.2.1 Considerations if using Red Hat .....	30
3.3 Downloading the code .....	31
3.4 Finding the documentation .....	34

3.5	Checking the Linux kernel settings . . . . .	35
3.6	Setting up the group and user ID for Oracle . . . . .	36
3.7	Set up your system for xWindows . . . . .	36
3.8	Getting the JDK . . . . .	37
3.9	Installing Oracle9i binaries . . . . .	38
3.10	Checking the environment. . . . .	38
3.11	Running the Universal Installer . . . . .	38
3.12	Setting up the .profile . . . . .	58
3.13	Connecting with a local client . . . . .	58
3.14	Connecting a remote client to the database . . . . .	59
	3.14.1 Updating tnsnames.ora . . . . .	59
	3.14.2 Starting the listener . . . . .	60
	3.14.3 Testing the connection . . . . .	60
3.15	Using the non-interactive install process. . . . .	61
3.16	Restarting the OUI. . . . .	62
3.17	Testing the de-install process . . . . .	63
<b>Chapter 4. Creating Linux virtual machines with shared Oracle binaries</b>		<b>69</b>
4.1	Overview . . . . .	70
4.2	Overview of the tests. . . . .	72
	4.2.1 Description of test . . . . .	72
	4.2.2 Hardware setup . . . . .	73
	4.2.3 Set up of software . . . . .	74
	4.2.4 Set up Linux . . . . .	80
	4.2.5 Installing Oracle9i . . . . .	85
	4.2.6 Oracle experiences . . . . .	85
4.3	Conclusion. . . . .	85
<b>Chapter 5. Migrating Oracle9i to Linux for zSeries</b>		<b>87</b>
5.1	The process for moving the database . . . . .	88
5.2	Overview . . . . .	88
	5.2.1 Memory restrictions. . . . .	88
	5.2.2 Database size: data files . . . . .	89
	5.2.3 Compatibility issues . . . . .	90
	5.2.4 Database functional issues . . . . .	90
5.3	Moving a small database. . . . .	91
5.4	Moving a large database. . . . .	93
<b>Chapter 6. Using RMAN to handle backup and recovery</b>		<b>103</b>
6.1	Introduction . . . . .	104
6.2	Overview of RMAN . . . . .	104
6.3	RMAN Media Management Interface . . . . .	106
6.4	Activating archivelog mode . . . . .	107
6.5	Using RMAN without a recovery catalog. . . . .	109

6.6	Using RMAN with a recovery catalog	110
6.7	Configuring RMAN	111
6.8	Creating and using backup scripts	111
6.9	Sample BACKUP operations	112
6.10	Sample RESTORE/RECOVER operations	114
6.11	Reporting functionalities of RMAN	116
<b>Chapter 7. Installing and configuring Oracle Enterprise Manager</b> . . . . . 119		
7.1	Documentation	120
7.2	Introduction to Intelligent Agents and OEM	120
7.3	Installing and configuring an OEM configuration	121
7.4	Installing and configuring an IA on Windows 2000	122
7.5	Installing the Oracle Enterprise Manager	127
7.6	Installing the Intelligent Agent and Data Gatherer	128
7.6.1	Introduction to Intelligent Agents and OEM	129
7.6.2	Installation of the Intelligent Agent	129
7.6.3	Starting the Intelligent Agent	129
<b>Chapter 8. Performance and scalability considerations</b> . . . . . 131		
8.1	Virtual machine scheduling by z/VM	132
8.1.1	Processor usage	136
8.1.2	Linux storage management and swapping	137
8.1.3	Hints for tuning Linux usage	139
<b>Chapter 9. Using Statspack to collect VMSTAT information</b> . . . . . 141		
9.1	Statspack	142
9.2	VMSTAT	142
9.3	Gathering VMSTAT as part of Statspack	143
9.3.1	Installing Statspack	143
9.3.2	The VMSTAT shell script	144
9.3.3	Procedure to run Statspack and shell script	146
9.4	Summary	148
<b>Chapter 10. Implementing RAC on Oracle9i on Linux on zSeries</b> . . . . . 149		
10.1	Preparation	151
10.1.1	z/VM setup	152
10.1.2	Linux setup	154
10.2	Oracle Cluster Manager install	159
10.2.1	Starting Cluster Manager	167
10.3	Oracle9i install	168
10.4	Database creation	170
<b>Chapter 11. Using WebSphere with Oracle9i</b> . . . . . 173		
11.1	Overview	174

11.2	Installing the repository database . . . . .	176
11.3	Configuration of the repository database . . . . .	177
11.4	Installing IBM HTTP server . . . . .	178
11.5	Installing WebSphere . . . . .	179
11.6	Running WebSphere as non-root user . . . . .	181
11.7	Starting the application server . . . . .	182
11.8	Testing the installation . . . . .	183
<b>Appendix A. Frequently asked questions about Oracle9i for zLinux . . .</b>		<b>185</b>
<b>Appendix B. Tools for VM performance data collection and analysis . .</b>		<b>189</b>
	z/VM Control Program (CP) . . . . .	190
	B.0.1 The CP INDICATE command . . . . .	190
	B.0.2 Accounting records . . . . .	191
	B.0.3 CP Monitor Facility . . . . .	192
	B.1 Optional programs . . . . .	193
	B.1.1 VM Performance Reporting Facility (VMPRF) . . . . .	193
	B.1.2 VM Real Time Monitor (RTM) . . . . .	194
	B.1.3 RMF PM . . . . .	195
	B.1.4 FCON/ESA . . . . .	195
<b>Appendix C. Examples of setting up the xWindows Interface . . . . .</b>		<b>197</b>
	Using a Linux Intel system as a client . . . . .	198
	Using VNC client and server . . . . .	198
	Using Exceed with Telnet . . . . .	199
<b>Appendix D. Linux Performance Data Collection and Analysis Tools . .</b>		<b>201</b>
	VMSTAT . . . . .	202
	XOSVIEW . . . . .	203
	TOP . . . . .	204
	IOSTAT & SAR . . . . .	205
	BMC PATROL . . . . .	206
	RMF PM with support for Linux enterprise server . . . . .	207
	ESALPS . . . . .	208
	Monitoring requirements . . . . .	208
	Standard interface . . . . .	208
	Performance Database (PDB) . . . . .	209
<b>Appendix E. Useful Linux information . . . . .</b>		<b>211</b>
	Useful vi command . . . . .	212
	Useful Linux commands . . . . .	213
	Linux directory structure . . . . .	214
	Important Linux configuration files . . . . .	214
	Tools to administrate Linux . . . . .	215

Putty .....	215
WinSCP.....	215
<b>Appendix F. Sample of response file.....</b>	<b>217</b>
<b>Abbreviations and acronyms .....</b>	<b>247</b>
<b>Related publications .....</b>	<b>249</b>
IBM Redbooks .....	249
Other resources .....	249
Referenced Web sites .....	249
How to get IBM Redbooks .....	250
IBM Redbooks collections.....	250
<b>Index .....</b>	<b>251</b>



# 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.

Information concerning Oracle's products was provided by Oracle. The material in this document has been produced by a joint effort between IBM and Oracle S/390 Specialists. This jointly produced material is copyrighted by both IBM and Oracle.

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 illustrates 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. 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 IBM's application programming interfaces.

## Trademarks

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

AIX®	OS/390®	VM/ESA®
CUA®	Perform™	WebSphere®
DB2®	PR/SM™	z/OS™
ECKD™	Processor Resource/Systems	z/VM™
e(logo)®	Manager™	zSeries™
 eServer™	PR/SM™	Lotus®
Hummingbird®	Redbooks (logo)™ 	1-2-3®
IBM®	Redbooks™	Notes®
IBM.COM™	RMF™	Word Pro®
MORE™	S/390®	
Multiprise®	SP™	
MVS™	Tivoli®	

The following terms are trademarks of other companies:

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

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

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

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

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

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

# Preface

This IBM Redbook describes experiences gained while installing and testing Oracle9i for Linux on zSeries, such as:

- ▶ Setting up the development systems at Oracle for the Linux on zSeries environment
- ▶ Installing the Oracle9i instances for Linux/390 on zSeries
- ▶ Performing basic monitoring and tuning exercises

This redbook is based on real-world experiences gained by team members and technical professionals during development and early testing of this product at:

- ▶ The IBM Oracle EMEA Joint Solution Center, Montpellier, France
- ▶ The IBM Oracle International Competency Center, San Mateo, California
- ▶ Early customer installations
- ▶ Development systems at Oracle in Redwood Shores, California

This redbook will be of use to those customers who are using Oracle9i for Linux/390 on zSeries for the first time.

## The team that wrote this redbook

This redbook 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 was an ERP Specialist at the ITSO in Poughkeepsie, New York.

**Denny Dutcavich** is a FTSS providing pre-sales technical support for Oracle Ebusiness Suite on IBM platforms, and the Oracle database for Linux on zSeries. Previously, he was a Software Engineer in the S/390 division, and provided technical marketing support for Oracle on OS/390.

**Bruce Frank** is a zSeries Oracle Specialist in the IBM/Oracle International Competency Center at IBM San Mateo.

**Michael Gorman** is a zSeries Oracle Specialist with Oracle Corporation, USA.

**Mark Jaenen** is a Certified Linux Specialist working for HSBC Trinkaus & Burkhardt in Germany. He has 5 years of experience with Linux. For the last year, he has been deeply involved in the implementation of Oracle on Linux on zSeries.

**Tom Russell** is a zSeries Specialist with IBM Canada. Presently, he is on a special assignment to Oracle in Redwood Shores, California.

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

Roy Costa  
International Technical Support Organization, IBM Poughkeepsie

Carlos Ordonez  
Eva Yan  
Dennis Wunder  
IBM Poughkeepsie

Chuck Morse  
Eugene Azuolas  
IBM Gaithersburg

Kirk Condon  
SBC Corp.

Betsie Spann  
Barry Perkins  
Bruce Olsen  
Marc Connolly  
Oracle Corporation

Thanks also to Maritza Marie Dubec, Terry Barthel, Alison Chandler, and Alfred Schwab for their editorial assistance; and Ella Buslovich for her graphics assistance.

## Notice

This publication is intended to help those who will be installing Oracle9i for Linux/390. The information in this publication is not intended as the specification of any programming interfaces that are provided by Linux/390 or Oracle9i for Linux/390. See the PUBLICATIONS section of the IBM Programming Announcement for Linux/390 for more information about what publications are considered to be product documentation.

## Comments welcome

Your comments are important to us!

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

- ▶ Use the online **Contact us** review redbook form found at:  
[ibm.com/redbooks](http://ibm.com/redbooks)
- ▶ Send your comments in an Internet note to:  
[redbook@us.ibm.com](mailto:redbook@us.ibm.com)
- ▶ Mail your comments to the address on page ii.





# Overview of Oracle9*i* on zSeries

Oracle has released Oracle9*i* Database Server for Linux on zSeries in August 2002. Several IBM sites have installed and tested the code. This document shares some of the experiences from these installations and the development environment used at Oracle in Redwood Shores.

This chapter is an overview of Linux on zSeries and Oracle solutions.

## 1.1 What is Linux

Linux is a UNIX-like operating system initially created by Linus Torvalds while a graduate student in 1991. The objective for developing Linux was to deliver a non proprietary operating system and application development environment, completely independent of underlying hardware architectures, which would offer maximum freedom to move applications from one hardware platform to another by simply recompiling code, without expensive, labor-intensive porting efforts.

Linux is a fully networked 32-bit/64-bit architecture, which supports multiple users, multitasking, and multiprocessors, with a user-friendly Xwindows graphical user interface. Continued development and testing of Linux is governed by the open source community, which uses the Internet as the primary vehicle for technical exchange. Linux source code can be downloaded free of charge from the Internet, and programmers are free to modify the code. However, the integrity of the official kernel source code is managed and maintained by a strict submission and review process controlled by the Linux Review Board, an international standards body for Linux.

Linux, like other open source software, is distributed under the terms of the GNU Public License (GPL), and is packaged, and distributed by approved distributors, such as Caldera, Red Hat, SuSE, and Turbo Linux. Distributor packages include the Linux operating system code that has been precompiled for specific hardware environments, along with other open source applications and middleware, such as Apache Web Server, SAMBA (file/print serving), Jabber (instant messaging), and IMAP/POP (mail servers). Distributors also offer Linux services and support packages, as does IBM Global Services.

## 1.2 What is Linux for zSeries

In 1998 IBM announced a commitment to support Linux on all its hardware platforms, including zSeries. The 31-bit version of the Linux operating is available from Linux distributors for S/390 G5 and G6 processors, and on 64-bit zSeries models. The Linux 64-bit support on the zSeries is available on open source and became available from SuSE in April 2001.

The Linux operating system has been running in a zSeries test environment since early 1999. As of December of 1999 (see IBM's Web site: <http://www.ibm.com/servers/eserver/zseries/solutions/s390da/linuxisv.html>) this site is continually updated to show all the Independent Software Vendors (ISVs) applications or infrastructure available for the zSeries platform.

Linux on zSeries is an ASCII environment that takes advantage of IBM zSeries hardware, especially for system availability and I/O performance. Of particular interest is the ability to run many Linux images under z/VM. This provides an excellent environment for server consolidation. It will more fully utilize system resources and facilitate system's management.

Linux is not a replacement for other IBM operating systems on zSeries or S/390, and will coexist with z/OS, OS/390, and VM/ESA. It supports such UNIX tools as sed, awk, and grep; compilers like C, C++, Fortran, Smalltalk, and Ada. Network tools like Telnet, FTP, ping, and traceroute are supported as well.

## 1.3 Why Oracle9i on Linux for zSeries?

The ability to combine the hardware characteristics of the IBM zSeries with the openness of Linux provides significant benefits to users.

IBM's zSeries has over the years demonstrated its unique ability to run multiple diverse work loads. This key strength also applies to Linux workloads. zSeries is particularly well suited to the hosting of multiple lightly to moderately loaded servers. (Examples of this type of server would be firewall servers, print and file serving, domain name servers, Internet news servers, or Web serving that is not processor intensive.) Most IT installations use multiple outboard servers to perform these functions. Consolidation of these functions onto a single zSeries can provide the following benefits:

- ▶ zSeries qualities of service: No other platform offers the qualities of service available in zSeries.
- ▶ Reduced hardware costs: Processors, storage, memory, etc. are now shared on a single Zseries.
- ▶ Reduced software costs: Software licenses spread over several machines and operating systems in most cases are reduced when consolidated onto Zseries processors.
- ▶ Reduced networking costs: Physical networking gear like routers and cabling are no longer necessary on a single Zseries. Servers communicate through inter-system facilities on Zseries (hardware or software).
- ▶ More efficient inter-server communication: Inter-server communication is faster than physical networking.
- ▶ Reduced systems management/support costs: Supporting multiple servers on a single platform requires less effort, fewer people.
- ▶ Reduced deployment time for new servers, since new servers are added virtually instead of physically, the time required to create a new server is minutes, not weeks.

Linux for Zseries offers many advantages to customers who rely upon zOS systems to run their businesses. While zOS has key strengths in the areas of data acquisition through high volume transaction processing and data management, its value is enhanced by the addition of Linux for Zseries in different ways.

Linux for Zseries brings the Zseries user the ability to access host data efficiently by using high speed, low latency, inter-partition communication. Benefit is derived from the elimination of outboard servers, routers, and other networking gear, reduced floor space, and reduced maintenance cost. Linux also has the well-earned reputation for rapid deployment of applications—giving Zseries users a key choice in how they choose to deploy an application.

Linux for Zseries takes advantage of the qualities of service in the Zseries hardware—making it a robust industrial strength Linux, while Zseries native services are available for applications that require the qualities of service inherent in zOS, as well as those of the Zseries hardware.

There are several other advantages to running Linux on zSeries. They are described in the following sections.

### **1.3.1 Expanded application portfolio**

The non proprietary environment of Linux opens the door for zSeries to perform as an application development and deployment server. The Linux application portfolio will greatly increase the number of applications available to zSeries customers who want to continue to leverage the critical advantages of the platform to run business applications on a highly available and reliable architecture. Customers will now have the flexibility to develop new applications directly on Linux for zSeries, and run them on any Linux supported platform; or to develop and test Linux on other platforms, such as the desktop, and then run the applications on Linux on zSeries, with a simple recompile. As applications from ISVs become generally available, customers will be able to deploy them quickly on zSeries without any special porting effort.

### **1.3.2 Cost savings**

A new feature designed specifically for the Linux operating environment, the Integrated Facility for Linux (IFL), is now available on G5, G6, and all zSeries server models including the Multiprise 3000. The IFL gives you the ability to dedicate processors to the Linux operating system on a logically partitioned machine, transparently to the z/OS operating system. Processors dedicated to the Linux LPAR under IFL are priced at a lower rate than those for the z/OS environment. The added capacity (because it is dedicated to Linux workloads) does not increase the software licensing fees of the zSeries environment.

Software pricing is confined to the capacity of only those processors enabled to the zSeries LPARs on the system. (This reflects pricing as of the publishing of this book, and may have changed since the publication date. Consult your IBM server sales representative to obtain the most current pricing structure information.)

### 1.3.3 Server consolidation and faster time to market

All of the great flexibility and openness of Linux, combined with the outstanding qualities of service of zSeries results in an industrial-strength Linux environment.

zSeries has several options for partitioning server resources into multiple logical servers. The PR/SM hardware feature provides the ability to partition the physical machine into as many as 15 logical servers, or LPARs, with dedicated, or shared, CPU and memory. The z/VM operating system allows you to partition an LPAR horizontally into multiple logical operating system images. These unique partitioning features provide the capability to quickly and easily consolidate a large number of Linux servers onto a single zSeries server. This industry-leading zSeries technology for dynamically sharing processing resources across multiple logical systems delivers value to the Linux environment not provided by any other architecture on the market today.

Figure 1-1 on page 5 illustrates the typical Linux or UNIX application deployment strategy used by most installations deploying applications on non-zSeries servers today.

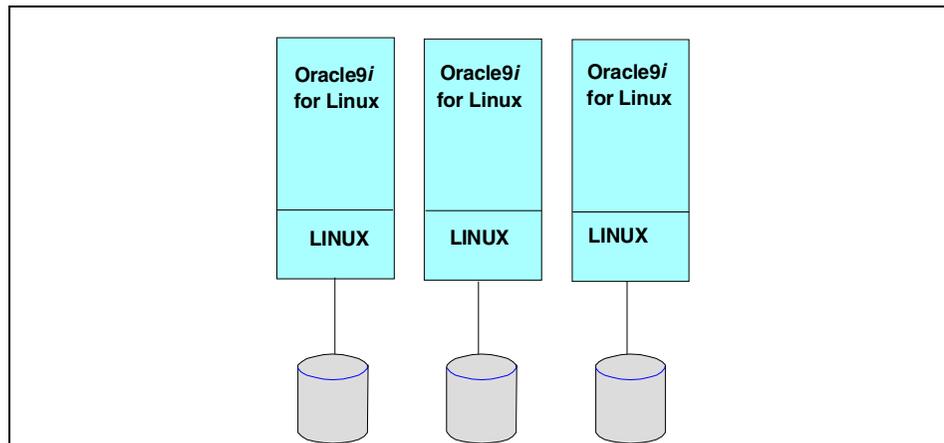
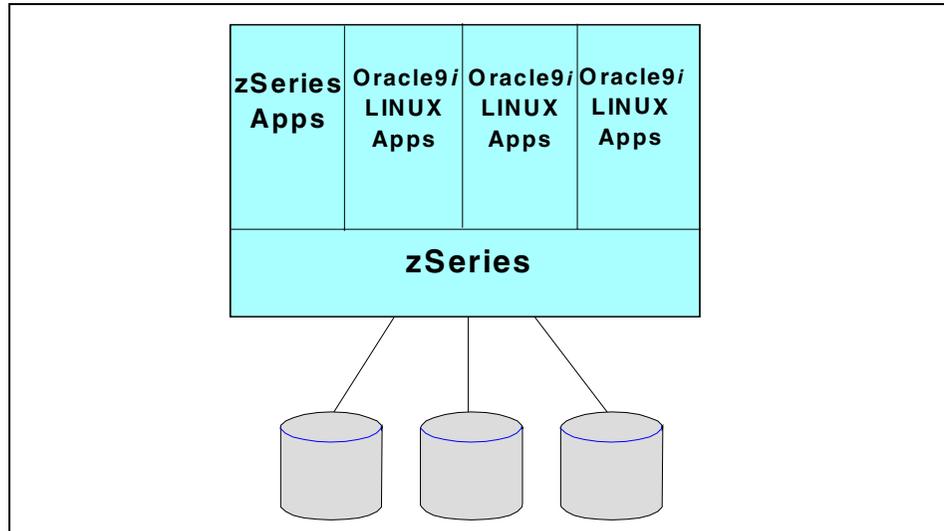


Figure 1-1 Linux application deployment on server farms

Without the robust operating environment provided by the z/OS operating system, and the industry-leading reliability of the zSeries hardware, most installations choose to deploy each new application in its own isolated operating system environment, on its own server, with a dedicated database or database partition. This leads to large server farms, with applications that cannot be easily integrated, and require complex systems management.

Contrast the configuration in Figure 1-1 to the flexibility available on the zSeries-S/390 architecture, illustrated in Figure 1-2.



*Figure 1-2 Linux application deployment on zSeries*

The unique capabilities of zSeries for running multiple operating images simultaneously and sharing processing resources dynamically, supports many diverse workloads and multiple applications on a single server, with outstanding interpretability and integration between applications. Database sharing with integrity and ease of systems management are additional unique benefits that zSeries brings to the Linux operating environment.

In fact, many customers are blending the data richness of the zSeries environments with the Web capability of Linux applications to deliver highly integrated, cost effective e-business solutions today.

## 1.4 What is Oracle9i for Linux on zSeries

Oracle has four main family of products:

- ▶ Oracle9i Database Server
- ▶ Oracle9i Application Server
- ▶ Oracle 11i EBusiness Suite
- ▶ Oracle Collaboration Suite

At the time of writing of this book, only the Oracle9i Database Server is available on zSeries with Linux. Click the following URL for the latest information:

<http://www.technet.oracle.com/support/metalink/content.htmlmetalink.oracle.com>

### 1.4.1 Oracle9i Database Server

Oracle delivered their production level Oracle9i Enterprise Edition Database Server on September 1, 2002. This release included the Oracle9i Release 2 (9.2) Enterprise Edition Server.

This release includes most of the components of the Enterprise Edition including:

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

The client package is also included, including Net8, OCI, and the Pro\*C/C++ precompiler.

The following functions are not yet provided:

- ▶ Pro\*COBOL - Planned for 9.2.0.2 patch set.
- ▶ OracleText does not have the INSO filters.
- ▶ The Oracle Management Server (OMS) function of OEM does not run in Linux.
- ▶ No generic connectivity (ODBC and OLE)
- ▶ Authentication with Verisign, CyberTrust, and Entrust

This provides a complete Oracle9i database for development, testing, and production in Linux for zSeries.

## 1.4.2 Oracle9i Application Server

In many ways, Oracle9i Application Server is well suited to Linux for zSeries. Linux, at its current state of development, is better suited for horizontally scaled workloads, such as print serving, Web serving, and simple application serving; or smaller scale database workloads. In building Oracle9i Application Server, Oracle uses many advanced elements for supporting Internet computing requirements like portals, security, commerce, etc. Some of these elements are not yet available outside the Intel and more established UNIX environments. Oracle and IBM are working to ensure that these partner products can be delivered.

## 1.4.3 The Oracle e-business Suite

Oracle has delivered its full suite of products, Oracle9i, Oracle9i Application Server, and the Oracle e-business Suite on Linux, on Intel hardware. As of this writing, Oracle has not made a public commitment to support the Oracle e-business Suite on Linux for zSeries. Oracle is interested in hearing from any customer who is interested in implementing Oracle e-business Suite on Oracle for Linux on zSeries.

## 1.4.4 The Oracle Collaboration Suite

The newest suite of products is the Oracle Collaboration Suite. The product 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. At the time of this writing, the Oracle Collaboration Suite is not supported in Linux for zSeries.

## 1.5 What distributions of Linux does Oracle support?

Today, Oracle supports Caldera, Red Hat, and SuSE distributions on Intel systems. Oracle9i for Linux on zSeries is intended for use on SLES7 (SuSE Linux Enterprise Server 7) and Red Hat 7.2. As of this writing, the certification for SuSE is complete and in process for Red Hat 7.2.

## 1.6 Obtaining a copy of Oracle9i for Linux on zSeries

The Oracle9i Database Server can be purchased through the Oracle Store at: <https://oraclestore.oracle.com/> or CD images can be downloaded from the Oracle Technical Network at: <http://technet.oracle.com/>

## 1.7 Sizing Workloads for Oracle9i on Linux for zSeries

The selection of an application to start testing the Oracle9i Database Server, or any application for that matter, is critical to the success of the test. It makes sense to select an application that has the characteristics of most of the applications that would be considered for server consolidation. It is very important to get a sizing of the workload to be moved to Linux for zSeries. It is also sensible to start with an application that is small or not complicated, and which uses that as a vehicle for skill building in this area.

It is important to understand that a *sizing* is an estimate of capacity needs. The performance team in Poughkeepsie developed a sizing methodology for equating a workload on a UNIX or Intel platform to its equivalent MIPS on zSeries. While this process has been generally accurate for several years, it is not a capacity plan. Capacity plans are based on the past performance of a system. The changing capacity needs over time provide an accurate way to predict future needs.

If this is a new application that has never run before, the sizing becomes a bit more difficult. Predictions can generally be made from testing done during development. If this is a vendor's application, they may be able to provide sizing assistance for their application in a Linux environment.

There are two significant reasons to start with a sizing. First it is important to understand the resources needed to run the application. This includes peak utilization on this server as well as when others peak. The sum of all peaks becomes important as more servers are moved to Linux for zSeries. In fact, this step helps start the process of understanding what the Total Cost of Ownership (TCO) results will be.

The second important reason for performing a sizing is to set expectations for testing of the workload that is moved. This will provide a starting point that can be used for tuning in either Oracle9i, Linux, or z/VM if necessary.

There is a service called SIZE390, which is available to provide pre-sale processor sizing estimates to IBM sales representatives and Business Partners for IBM zSeries and S/390 systems running z/OS or OS/390 and zLinux. It is now available world wide.

SIZE390 provides a questionnaire that requests all the information needed to size workloads. The questionnaire can be obtained by IBM sales representatives or Business Partners through TechXpress.

You should be prepared to supply the following:

- ▶ Machine model and characteristics such as:
  - CPU MHz
  - Number of CPUs
  - Memory
- ▶ Type of application:
  - DSS
  - OLTP
  - DNS
  - etc.
- ▶ Approximate number of users
- ▶ Utilization profile:
  - VMSTATS should be provided for a peak period.

This sizing deliverable provides some information about sizing estimates for Linux. Keep in mind, to get exact sizing estimates you will need to test in each customer's environment, but this data is useful for measuring other possible consolidation efforts in the customer location.



## How to deploy Oracle9i on Linux/390

This chapter discusses the decisions we made when we defined the Linux environment at Oracle. The Linux machines were used to build the Oracle 9i Release 2 (31-bit) for Linux/390 product at Oracle headquarters. The options available for deploying this many Linux machines are discussed, and the decisions we made are outlined.

## 2.1 Deployment options

The major decisions to be made when deploying Linux/390 are:

- ▶ Should the Linux machine images be guests under z/VM, or should they be run natively in a logical partition provided by the Processor Resource/Systems Manager (PR/SM) feature available on zSeries processors?
- ▶ Should the Linux images be run on the zSeries processor general purpose CPs, or using IFL CPs?
- ▶ Should we use the SuSE or Red Hat distribution of Linux?
- ▶ What network topology should we use to connect the Linux machine images to the network?
- ▶ How much CPU and memory resource should be allocated to the Linux machines?

### 2.1.1 z/VM or LPAR?

zSeries processors have a feature called Processor Resource/Systems Manager (PR/SM) which allows the z/800 or z/900 processor to run in Logical Partitioned (LPAR) mode. In this mode, the zSeries processor can be divided up into up to 15 logical partitions (LPs). Each of these LPs can run a separate operating system, such as z/OS, z/VM or Linux. The PR/SM feature isolates the different LPs so they run completely independent from one another.

z/VM is an operating system that provides support for multiple *virtual machines*, each with their own operating system. The virtualization capability of z/VM allows you to run an effectively unlimited number of operating system images on the same z/900 or z/800 machine. By providing virtual machine facilities, z/VM also simplifies the management of multiple guest operating systems. See *z/VM General Information*, GC24-5991 for more information on z/VM. The redbook *Large Scale Linux Deployment*, SG24-6824 also has a chapter that provides a basic introduction to z/VM.

#### Deploying in a logical partition

LPAR mode supports a maximum of 15 logical partitions. Each of these LPs can run a different operating system. Many customers will already run their zSeries processor in LPAR mode to support z/OS or OS/390 images for their normal work. It is not that difficult to add another logical partition to run Linux.

The definition of the hardware configuration of the machine is done by the customers' system staff using a tool called the *Hardware Configuration Dialog* (HCD). HCD creates an I/O definition file (IODF) containing the definition of all the LPs in the machine, and the I/O devices and channels connected to the LPs. This configuration is *activated* by a process called *Power-on Reset*, which completely initializes the hardware system, requiring an outage to any work running in any logical partition.

Each logical partition is defined in an LPAR Image Profile. This profile is maintained by customer operations or systems programming staff using the zSeries Hardware Management Console (HMC). This profile specifies the number of CPs used by the logical partition and whether these CPs are dedicated to the LP or shared with other LPs. It also specifies the amount of storage to be assigned when the LP is activated, and other attributes. Note that all storage for the Linux image is real storage, that is, it is not shared with any other LP. These operations are discussed in *Hardware Management Console Operations Guide*, SC28-6815, and are outside the scope of this redbook. Suffice it to say that configuration changes to any Linux images running directly in a logical partition will require careful planning, and the changes must be implemented by systems programming or operations staff.

The LPAR dispatcher is part of the Processor Resource/Systems Manager hardware feature. This dispatcher ensures that the work running in the CPs used by the various logical partitions get control in an appropriate manner. LPs can share the real CPs or they can be defined as having dedicated CPs. Each LP is assigned an *LPAR weight* in the image profile. The dispatcher uses this weight to manage the access of the shared CPs to time executing on the real CPs. If you notice that a workload in a given logical partition is suffering from a shortage of processor time, then one possible remedy action might be to increase the weight of the LP, ensuring more access to the real processors. Changing these weights will effect the balance of the work running on all the LPs, and so is usually done by operations or system programming staff, and must be carefully planned.

The device numbers used to address the I/O devices used by the Linux machine will be the real device numbers of the devices used. That is, they are assigned by the customer hardware planning staff, and cannot be changed arbitrarily. If multiple Linux images are run in multiple logical partitions, then they will assuredly have unique addresses. Each Linux machine will have to be customized separately by specifying the unique addresses for the disk and network devices.

## Deploying under z/VM

z/VM presents a unique approach to computer operating systems. It provides each end user with an individual working environment known as a virtual machine. The virtual machine simulates the existence of a dedicated real machine, including processor functions, storage, and input/output (I/O) resources.

z/VM supports an (effectively) unlimited number of virtual machines. Each of these virtual machines is defined in a directory which specifies, by user ID, the characteristics of the virtual machine that will be provided when a user logs on with that specific user ID. Permanent changes to the virtual machine configuration are made by changing the directory entry and logging off and on again. Temporary additions to the virtual machine can be made by executing z/VM commands. Changes to the VM directory are usually done by the VM systems programmer, without involving operations staff.

The directory specifies the number of CPs to be used by the virtual machine. These CPs are dispatched on the real CPs by the z/VM dispatcher. It is theoretically possible to define more CPs in the virtual machine than are really installed on the real machine, but this is rarely done except for testing. The directory also specifies the amount of storage to be assigned when the virtual machine is logged on. Note that all storage for the Linux image is virtual storage, that is, it is pageable storage provided by z/VM. The sum of storage for all the Linux virtual machines can exceed the total real storage available to z/VM. This is called *overcommitment*. A severe overcommitment may cause a performance problem as z/VM moves the virtual machines pages between storage and the DASD page data sets.

The z/VM dispatcher ensures that the work running in the CPs used by the various virtual machines get control in an appropriate manner. Each virtual CP is dispatched on a real CP based on z/VM scheduling algorithms, which are discussed in more detail in Chapter 8., “Performance and scalability considerations” on page 131.

The device numbers used to address the I/O devices used by the Linux virtual machine are virtual device numbers assigned by z/VM based on information in the machines directory entry. If multiple Linux images are run in multiple virtual machines, it is strongly recommended that each Linux machine be given the same addresses. By providing each Linux guest machine with as close to identical directory entries, the configuration and operation of the Linux guests is significantly simplified. By mapping the unique real device numbers to identical virtual device numbers, each Linux machine need not be customized separately. The process of configuring a number of Linux machines with identical configurations is called *cloning*. Cloning is discussed in more detail in Chapter 4., “Creating Linux virtual machines with shared Oracle binaries” on page 69.

## 2.1.2 CPs or IFLs?

The central processing unit of a zSeries machine is called a Central Processor (CP). (Sometimes we refer to a CP as an *engine*, especially when talking about multi-processor, or multi-engine machines.) The logical partition where either the Linux machine or z/VM runs can be specified use either general purpose CPs or Integrated Facility for Linux CPs. The Integrated Facility for Linux (IFL) is a feature on the z/900 or z/800 processor that dedicates a processor for Linux (or z/VM) images. Ordering this feature means you can have additional processing capability exclusively for Linux workloads without impacting the z900 model designation, and without increasing charges for zSeries software on the rest of the server.

Because adding the IFL CPs is less expensive than adding general purpose CPs, we recommend that you use IFL engines for deploying the Linux machines.

## 2.1.3 SuSE or Red Hat?

At the time of this writing, the SLES 7 distribution from SuSE is the only distribution that includes the drivers for operating the OSA/Express card in Queued Direct Input Output (QDIO) mode. These drivers are distributed by IBM as object-code-only (OCO). Using the Open Systems Adapter (OSA) card in this manner is much more efficient, as the amount of CPU time used per TCP/IP packet transferred is much less than non-QDIO mode. We recommend you use QDIO mode.

The Red Hat 7.2 distribution does not distribute the IBM OCO drivers. If you deploy with this distribution, you have to obtain the latest OCO drivers from the IBM Developerworks site:

[http://www.ibm.com/developerworks/opensource/linux390/special\\_oco\\_rh\\_2.4.shtml](http://www.ibm.com/developerworks/opensource/linux390/special_oco_rh_2.4.shtml)

The install process is described in the redbook *Large Scale Linux Deployment*, SG24-6824.

As well, the SuSE distribution includes the Logical Volume Manager (LVM). See “Disk capacity and logical volumes” on page 22 for a discussion of the Logical Volume Manager. We recommend that you use Logical Volume Manager to improve the performance of the DASD system.

At the time of this writing, Oracle 9i Release 2 (31-bit) for Linux/390 was only certified by Oracle Corporation on the SuSE SLES 7 distribution, and so that is what we used in our tests. Certification on a Red Hat distribution is planned and is in progress.

## 2.1.4 Network topology

One of the important decisions that must be made when deploying Linux machines is the network topology. Many Linux machines are configured with private TCP/IP addresses and exist behind a firewall. Other Linux machines are configured to run as peers of the other operating systems in the customer's environment.

### Multiple logical partitions on zSeries

Here are three figures that show some options on how to connect the Linux guest machines to the network.

In Figure 2-1 on page 17 we show connecting the Linux machines to a different TCP/IP subnet than the z/OS logical partitions on the same zSeries processor. The recommended way to connect multiple logical partitions together is by the use of HiperSockets. HiperSockets provides very fast data movement between logical partitions. It provides up to four internal LANs accessible by all logical partitions on the zSeries server. There can be up to 1,024 devices (TCP/IP images) across all four HiperSockets LANS, supporting up to 4,000 IP addresses. It achieves its high speed by using a memory-to-memory move technique that is similar to cross-address-space memory moves. These moves use the memory bus, but it does not use the CPU cache, and thus it has very little effect on other memory activity. Much more information on how to implement HiperSockets is included in the redbook *zSeries Hipersockets*, SG24-6816.

The recommended technique to connect multiple Linux virtual machines together is the use of a z/VM 4.3 feature called *Guest Lan*. This is like a virtual HiperSockets connection. Introduced with z/VM Version 4.3, the QDIO VM Guest LAN simulates a network interfaced via a QDIO-mode adapter such as the OSA-Express Fast Ethernet. As far as the guest operating system is concerned, the QDIO VM Guest LAN acts as an Ethernet network with functions such as broadcast and multicast, which are supported. A z/VM Guest LAN can only support connections to virtual machines on the same z/VM system.

To use a VM Guest LAN, you need to:

1. Create a VM Guest LAN segment to act as the network between virtual machines.
2. On each virtual machine to be connected to the VM Guest LAN, create a simulated network interface card (NIC). Each virtual machine will have the simulated NIC at the same virtual address. This simplifies the cloning of the configuration.
3. Attach each virtual machine's simulated NIC to VM Guest LAN segment.
4. For each Linux guest to be connected to the VM Guest LAN segment:

- a. Start the Linux guest.
- b. Configure the Linux network interface for the VM Guest LAN.

More detailed information on implementing a guest LAN is contained in the redbook *Large Scale Linux Deployment*, SG24-6824.

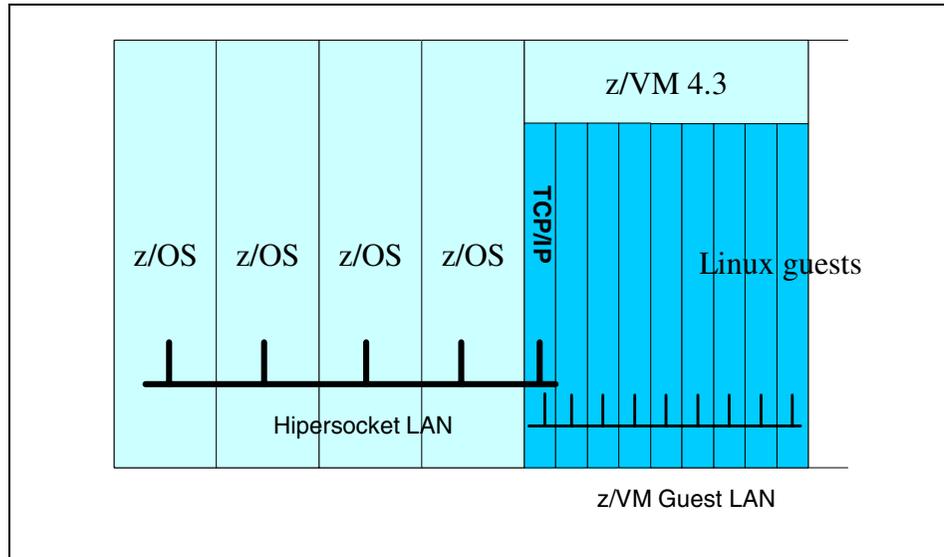


Figure 2-1 HiperSockets and z/VM Guest LAN

The HiperSockets LAN and the z/VM Guest LAN are obviously on different TCP/IP subnets. The TCP/IP stack in the z/VM logical partition acts as a router to connect the two subnets together.

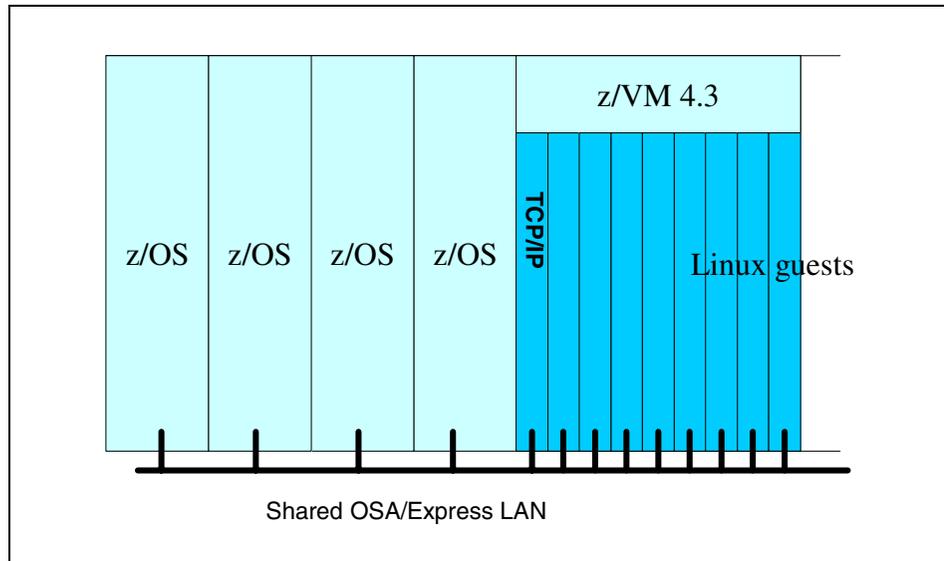


Figure 2-2 Peer connection through shared OSA/Express

Another technique (illustrated in Figure 2-2) which is almost as fast as a HiperSockets connection, is to share the OSA/Express adapter between all the logical partitions. An OSA/Express adapter is the standard adapter card used for connecting a zSeries processor to a network. When running in Queued Direct Input Output (QDIO) mode, the card is extremely efficient. Each port on the adapter card can support up to 80 unique channel connections from unique logical partitions or virtual machines. Data transfer between these users does not involve the real network at all. The data packets are sent from one address to another directly, and are buffered in the storage in the adapter card. This transfer is almost as fast as using HiperSockets, which does not use an adapter card at all.

The OSA/Express card supports Fast Ethernet (10 or 100 Mb), Gigabit Ethernet (1000 Mb), 155 ATM, or Token Ring (4, 16, or 100 Mb) connections. In QDIO mode, each port takes three device addresses. The adapter card and TCP/IP share a common storage area for memory-to-memory communication, reducing system overhead and improving performance. Data can move directly from the OSA-Express microprocessor to system memory. There are no read or write channel programs for data exchange. For write processing, no I/O interrupts have to be handled. For read processing, the number of I/O interrupts is minimized.

More information on implementing an OSA/Express card can be found in the redbook *OSA-Express Implementation Guide*, SG24-5948.

By sharing the OSA/Express adapter card, all the Linux machines are on the same TCP/IP subnet as the other logical partitions.

### **Channel-to-channel connections**

A common technique for connecting virtual machines to a network is to use virtual channel-to-channel (CTC) devices. An example of this is illustrated in Figure 2-3. A CTC device connects two machines in a point to point fashion. z/VM allows you to define an effectively unlimited number of *virtual* CTC adapters. Define a CTC adapter in the Linux machines directory entry, a corresponding CTC adapter in the TCP/IP machines directory, and use the z/VM COUPLE command to make the connection. TCP/IP in both z/VM and Linux can be configured to use the CTC connection for TCP/IP traffic.

Although the use of CTCs is common, we have found that Guest LAN is easier to manage and provides better performance, so we recommend that you use the Guest LAN facilities of z/VM Version 4.3.

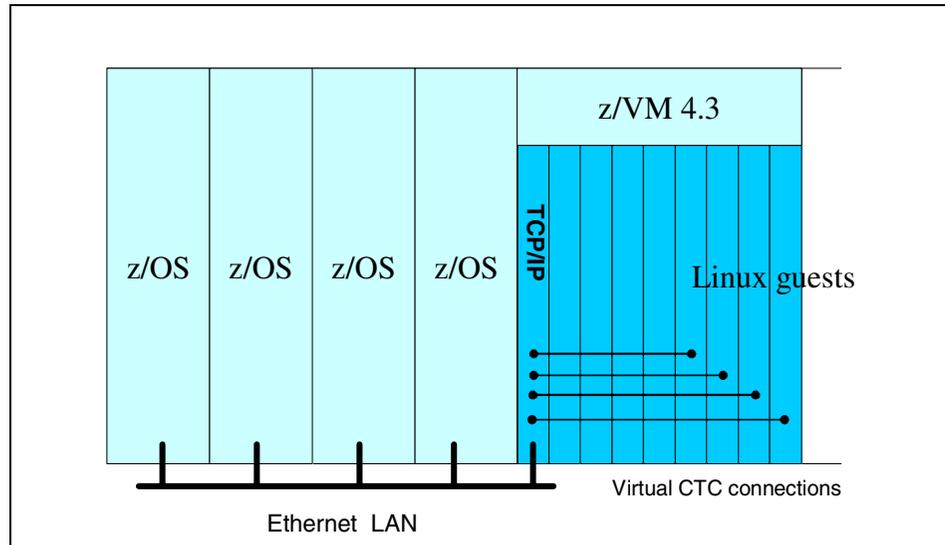


Figure 2-3 Peer Connection using Virtual Channel to Channel (CTC) adapters

### **IUCV connections**

The Inter-User Communications Vehicle (IUCV) is a communications facility that allows a program running in a virtual machine to communicate with other virtual machines. It is provided as a standard facility of z/VM. It provides a point-to-point connection between two virtual machines. The z/VM TCP/IP supports communication over this path, and there are drivers for IUCV connections standard in the Linux distributions for zSeries.

The topology of a network using IUCV is the same as that illustrated in Figure 2-2 on page 18 using CTC adapters.

We have found that Guest LAN is easier to manage and provides better performance, so we recommend that you use the Guest LAN facilities of z/VM Version 4.3 rather than point-to-point connections with IUCV.

### Network Topology without z/VM

If you deploy a number of Linux machines in logical partitions, without the use of z/VM, then each Linux machine will be a peer on the network with the other operating systems running in other logical partitions. This is illustrated in Figure 2-4. Note that unlike a virtual machine under z/VM where the network adapter address (NIC) has the same address as other virtual machines, when you deploy in a logical partition, the operating system must see the real device address of the NIC. Each of the Linux machines in Figure 2-4 will use different device numbers to access the OSA/Express adapter.

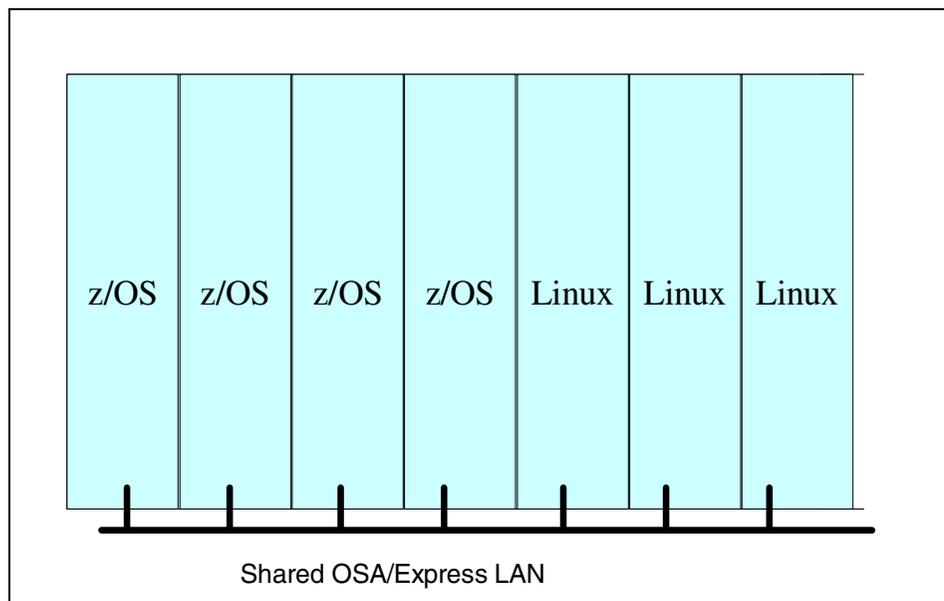


Figure 2-4 Peer Linux guests in logical partitions

## 2.1.5 Capacity planning

Deploying Linux on zSeries processors must be approached with the same methodology as deploying any other operating system. Consolidating multiple NT or Linux servers onto Linux for zSeries can be a great idea, and can save money. However, if you are planning to consolidate servers that are doing existing

workloads, then you must measure the utilization of the existing servers and also do some calculations to ensure that the configuration of Linux on the zSeries processor is appropriate. The ITSO Redpaper *Server consolidation with Linux for zSeries*, REDP0222, covers many of the capacity planning issues.

Performance considerations for Linux on zSeries are discussed more completely in Chapter 8., “Performance and scalability considerations” on page 131. This section will only discuss the basic considerations.

## Number of CPs

The speed and capacity of a z/800 or z/900 machine are well understood. However, deploying Linux on these machines will probably share these resources through the use of LPAR with other workloads and other operating systems. It will be fairly rare to dedicate a complete zSeries machine to a single copy of Linux, even though that is the norm in other platforms. Each of the small servers that are being consolidated onto the zSeries processor has the entire capacity of the server available for peak loads, but probably (hopefully?) runs at an average utilization much less than 100%. On the zSeries machine, all resources are shared, and hopefully the peak activity of each Linux machine will not occur simultaneously. You must analyze each of these servers and calculate the expected utilization and number of CPs required to provide the capacity required.

There are tools available from IBM to assist in this process. For a detailed sizing study using a tool called SIZE390, contact your IBM representative. This tool can help analyze the utilization of the servers that will be consolidated onto the zSeries machine.

## LPAR weights

As discussed in “Deploying in a logical partition” on page 12, each logical partition is assigned a *weight*. Processor resources are given to the various logical partitions in the ratio of the weights. Normally, we recommend that the sum of the weights assigned to the logical partitions sum to 100 or 1000. If this is done, then the weight can be directly understood to be the percentage of total capacity of the zSeries processor that is given to the specific logical partition. Capacity greater than this percentage can be delivered to a given logical partition, but only if no other logical partition wants the resources. The weight, expressed as a percentage is the *target capacity* of the logical partition. If, for example, you want the Linux machines to get 40% of the capacity of the zSeries processor, then the sum of the LPAR weights for the Linux logical partition should be 40% of the total weights for all logical partitions.

The LPAR dispatcher keeps two pools of processors. A given logical partition is assigned a number of logical CPs that can come from the pool of regular shared CPs, or the IFL CPs. IFL processors are cheaper than general purpose CPs, and are therefore recommended. One consideration is that IFL CPs can only run Linux or z/VM, and the workload of the Linux configuration cannot automatically expand into the general purpose CPs. The two pools of CPs are kept separate by the LPAR dispatcher.

### **Shared or dedicated CPs**

When you define a logical partition you must specify whether the CPs used will be *dedicated* to the logical partition, or *shared* by all the logical partitions on the processor. Dedicating the processor is normally done if the workload is critical and requires a guaranteed resource. For example, if you were running z/VM in a number of IFL CPs, you would probably dedicate the IFL CPs. Sharing is recommended when the Linux machine requires less than a full CP in capacity. Running Linux in a logical partition without z/VM would normally be done by sharing the CPs. LPAR weights only apply to shared CPs.

### **Memory capacity**

Deciding how much memory to assign to the Linux image is non-trivial. Linux uses all the memory in the machine (whether virtual machine or logical partition) as disk buffers. As processes request storage, some buffers will be freed to satisfy the storage request. The result of this is that the monitors, such as TOP, will always report that the storage utilization is close to 100%. It is difficult to determine the minimum storage requirement.

If you are deploying in a logical partition, changing the storage size requires the use of the HMC to change the LPAR image profile. If you are deploying under z/VM, then the virtual machines directory entry must be changed. Note that if you are running several Linux guests under z/VM that the total memory requirement may be quite high. This implies that you may require several disk volumes for z/VM to use for paging. If you are severely overcommitted, then the paging rates from z/VM may be too high to provide adequate performance for the Linux machines.

### **Disk capacity and logical volumes**

Disk storage on a zSeries processor is most commonly a 3390 Model 3 or Model 9. The model number is approximately the capacity of the disk volume in gigabytes. Even 9 GB is not big enough for many Linux file systems. As well, Linux will only start one I/O operation per physical volume, even if there are multiple paths to the volume. Because of this, we recommend that you combine several physical volumes together to create a striped logical volume using the Logical Volume Manager (LVM) facility of SuSE Linux.

To define a logical volume, there are three steps:

1. We have to define and initialize the *physical volumes* (PVs) that we plan to use
2. Define *volume groups* where we group the PVs together that we want to manage in one group.
3. Next, we set up logical volumes on each volume group.

We recommend that all the physical volumes that you include in the logical volume be the same size. This makes it possible to *stripe* the logical volume.

In Figure 2-5 there is picture representing a logical volume that was defined as not being striped (or as having a single stripe if you would prefer). Note that as data is added to the logical volume, the physical volumes in the volume group are filled consecutively. It is possible that data would be retrieved by Linux from more than a single physical volume at a time, but this would be relatively rare. One case where this occurs would be if one process was reading data located near the beginning of the file system, while another was processing data near the end.

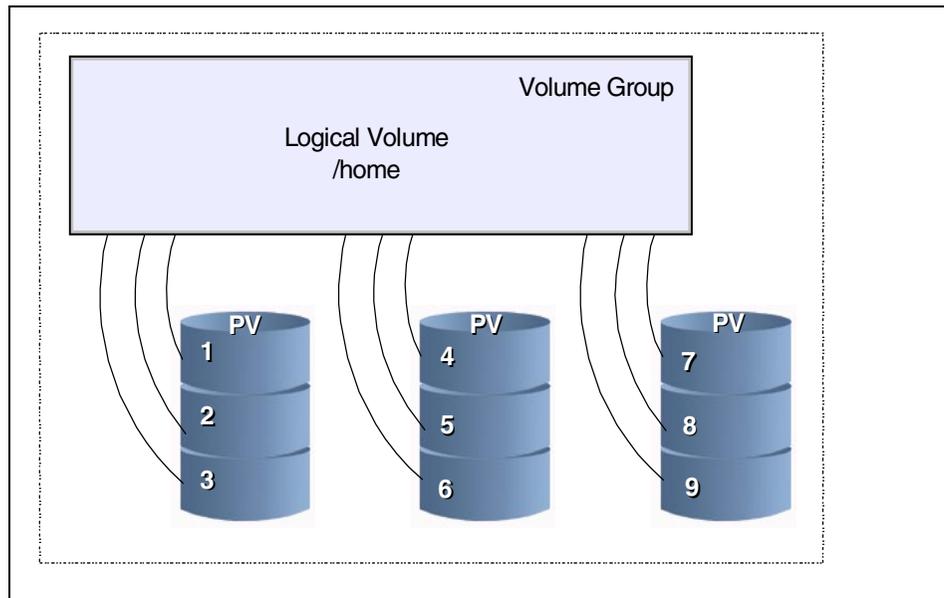


Figure 2-5 Non-striped logical volume

It is much more likely to get multiple I/O operations started simultaneously if the logical volume is defined as in Figure 2-6. In this case the data is placed in all the physical volumes in a round-robin fashion. Most files will be written into multiple physical volumes. I/O operations to a single file can be processed simultaneously as pieces of the file are on different physical volumes. This increases the performance of the Linux machine significantly, as it ensures that the multiple channel paths to the real devices are used.

The numbers in Figure 2-5 on page 23, and Figure 2-6 represent the order that data is placed in the logical volume. Note that in the non-striped logical volume, the data fills each physical volume before using the next physical volume.

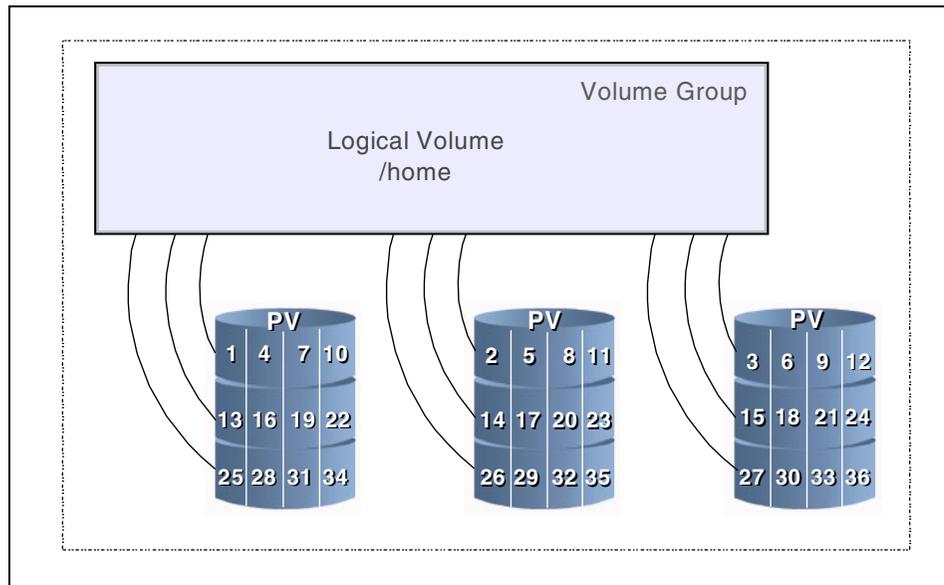


Figure 2-6 Striped logical volume

An unstriped logical volume can be extended by adding more physical volumes in a non-disruptive fashion. Extending the logical volume in this way is not possible with a striped logical volume. Even so, the performance advantages of a striped logical volume are significant, and so we recommend that you stripe the logical volumes.

More information about LVM can be found at the Sistina Web site at:  
[http://www.sistina.com/products\\_lvm.htm](http://www.sistina.com/products_lvm.htm)

There is also a chapter discussing LVM in the redbook *Linux for IBM @server zSeries and S/390: Distributions*, SG24-6264.

## Adding capacity to a Linux machine

With Linux on an Intel server, the server is a fixed size. Upgrading the capacity of the box means installing a new one, and moving the application.

With Linux on zSeries, more capacity can be added to a Linux server in many ways:

- ▶ Increasing the LPAR weight of the logical partition where Linux or z/VM is running will result in the logical partition getting more CPU resource from the LPAR dispatcher.
- ▶ Extra disk devices and storage can be added to the directory of the Linux machine at any time. Rebooting the Linux image will see the increase in resources.
- ▶ Unique facilities of zSeries can be used in ways unavailable to Linux on Intel servers. For example, sharing disk files between two Intel Linux servers would probably use Network File System (NFS) to connect the two servers. With zSeries, shared disk files are routine. With Linux on zSeries, read-only file systems can be directly shared. The performance of a shared disk subsystem is significantly better than network access through NFS.

**Note:** It is currently only possible to have all the Linux machines have the filesystem mounted as read-only. If updates must be made to the filesystem, then all the read-only users must unmount the filesystem, or be stopped while another Linux machine mounts the filesystem in read-write mode and makes the changes.

After the updates are done, the file system must be unmounted by the read-write user. The filesystem can then be mounted read-only by all the other users and shared processing can be resumed.

### 2.1.6 Customization of the SuSE distribution

We used the SuSE SLES 7 distribution with very few changes.

#### Rexec

In `/etc/inetd.conf` we uncommented the line containing the characters `in.rexecd`. This enables the rexec function which allows us to execute commands remotely. We used this for shutdown and startup automation.

We uncommented the line in `/etc/inetd.conf` that contains the characters `in.ftpd`. This enabled the FTP daemon, so we could send files to and from these machines. Because our machines were on a secure network, we also removed a security control. There is a file `/etc/ftpusers` which by default disallows the user `root` from using FTP. We removed the entry for `root` from this file.

## Patches

The SLES 7 distribution has a Linux Kernel of 2.4.7. We installed most of the patches supplied by SuSE on their first patch CD. This CD is labeled SLES-7-PatchCD-1-s390-20020522. This includes the so-called timer patch, which reduces the overhead of an idle Linux system. Also, the `raw` command, which is required for the Oracle Real Application Clusters (RAC) feature, did not work until the two patches `devs-2001.10.11-8.s390.rpm` and `rawio-2-33.s390.rpm` were applied. These are on the patch CD in the `ap2` directory.

## Dasd configuration

Each Linux machine was configured with a 800 cylinder disk for the root file system. A 3000 cylinder file was defined for the `/usr` files system, and a 1000 cylinder disk was used for the `/tmp` filesystem. As mentioned in 8.1.2, “Linux storage management and swapping” on page 137, we used two VDISK devices for swapping. This system was built and installed, and was used as a base for cloning all other systems.

All application files unique to each image were defined on striped logical volumes varying in size from 90 to 150 GB in size.



## Installing Oracle9*i* Database on zSeries with Linux

Oracle released Oracle9*i* for Linux/390 in August 2002. Several IBM sites have downloaded the code from OTN, installed, and tested the code. This chapter is based on the installation experiences from these installations.

Oracle9*i* for Linux/390 (31-bit) can be set up using a Linux guest under VM on an LPAR on zSeries or a S/390 or directly on a LINUX LPAR.

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

## 3.1 The system environment

We did our testing on four Linux guests on a Z900 at IBM Poughkeepsie, and four Linux guests on an IBM S/390 Mutiprise 3000 system at IBM San Mateo. This chapter is based on a Linux guest names LINUX1. The filesystem we used was a 6 GB logical file mounted as /oracle1.

## 3.2 The Linux environment

Have your systems administrator set up the LPAR with the resources you need such as CPU, memory, disk, and network connectivity.

To install Linux on a LPAR we recommend following redbook: *Linux for IBM zSeries and S/390: Distributions*, SG24-6264, 3.2., “Prepare the LPAR for Linux” then continue with Chapter 5., “Installing SuSE Linux.”

Here are the steps to install Linux:

1. Boot from tape, CD-ROM or FTP server.
2. Respond to the questions to setup the network:

```
=
=
== Welcome to SuSE Linux S/390 7.2 ==
=
=
First, select the type of your network device:
0) no network
1) OSA Token Ring
2) OSA Ethernet
3) OSA-Express Gigabit Ethernet (experimental)
4) Channel To Channel
5) Escon
6) IUCV (experimental)
Enter your choice (1-6):
```

Our choice for our native LPAR installation was number 2.

See the redbook *Linux for IBM zSeries and S/390: Distributions*, SG24-6264, 5.2., “Installing SuSE Linux on network setup.”

After entering the network definitions, a root password is requested.

The boot process does network checking, completes the boot process, and then the Linux prompt will appear. At this point you can Telnet from another machine in order to get a better terminal interface to continue with the install.

### **Definitions used for VM**

This is the Linux definition we used under VM at the ITSO in Poughkeepsie:

The setup is SuSE 2.4.7 guest Linux system (31-bit), running under z/VM 4.2 on z/900 LPAR:

- ▶ Storage size is 512M
- ▶ IP address is 9.x.x.x
- ▶ System name is VMLINUX2
- ▶ Root password is xxxx

Each system has these file spaces:

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/dasdb1	2.1G	73M	1.9G	4%	/
/dev/dasdc1	2.3G	2.0G	153M	93%	/usr
/dev/dasdd1	2.3G	2.0G	222M	90%	/opt
/dev/grp1/vol1	6.8G	20k	6.4G	1%	/oracle1
	this is 3 3390-3 vols in LVM				
/dev/dasdh1	2.3G	20k	2.1G	1%	/oracle2

**Note:** You can have *n* number of ORACLE\_HOME, but we recommend only having one Oracle instance per VM/Linux partition.

### **Linux variables**

There are entries to set for variables such as shared memory. See *Oracle9i Release Notes for Linux/390 Systems*.

### **Level of SuSE**

This is only supported on SuSE 7.2. It does not work on earlier versions.

### **GID for user ID**

The Oracle user ID that created the tar file had a GID of 500. The **tar** command will re-assign the new group ID of the user that does the un-tar. In one case we used a GID of 55, and in a second case we used a GID of 500. It did not seem to affect the installation.

### **Type of filesystem**

We used file type EXT2.

## **Documentation for Linux/390**

For reference about Linux/390 we used the following publications:

- ▶ *Linux for IBM e-server zSeries and S/390: Distributions*, SG24-6264
- ▶ *Linux for S/390*, SG24-4987-00
- ▶ *Linux on IBM eserver zSeries and S/390: ISP/ASP Solutions*, SG24-6299-00
- ▶ *Building Linux Systems Under IBM VM*, REDP0120

### **3.2.1 Considerations if using Red Hat**

This section notes the differences encountered when installing on Red Hat 7.2 (31-bit).

We followed the chapter about Red Hat in the redbook, *Linux for IBM zSeries and S/390: Distributions*, SG24-6264:

1. There are several methods to use to start the installation process.

For the installation process we used the method of copying the two CD-Roms to hard drive.

The other methods would have required us to set up a URL on a Web server to use the FTP method, as we did in SuSE. Also, when we booted from the CD-ROM directly to do the setup of Linux in the RAM, we did not set up my network correctly.

2. When you are installing, make sure to select developer tools and utilities because the C compiler and other libraries are required for the **make** command to work.
3. Note that we found that the Red Hat distribution uses more space than SuSE, if you select everything.
4. Setting up a large filesystem:  
Red Hat does not support Logical Volume Manager (LVM), so we used software raid to create a large disk, as the installation needs 2.75 GB, which is larger than a mod3 DASD.

The VM server VM directory has three minidisks for filesystems. Three minidisks (300-302) of 1000 cylinders each correspond to md0.

300 is /dev/dasdf, 301 is /dev/dasdg, 302 is /dev/dasdh.

This is our software raid file:

```
vmlinuxg2: espann > cat /etc/raidtab
```

```
raiddev    /dev/md0
raid-level  0
nr-raid-disks 3
chunk-size 4
persistent-superblock 1
```

```
device /dev/dasdf1
raid-disk 0
device /dev/dasdg1
raid-disk 1
device /dev/dasdh1
raid-disk 2
```

To create a raid volume issue the command: `/sbin/mkraid /dev/md0 md1`

To create a filesystem issue the command: `/sbin/mke2fs -b 4096 -R stride=3 /dev/md0`

Notice that the "-R stride=x" is the striping and is usually equal to the number of disks.

If the filesystem is shared between servers, such as a raw cluster, issue the following command to start it on the second server:

```
/sbin/raidstart /dev/md0
```

For more information, see the *Red Hat Linux 7.2 Customization Guide* at: <http://www.redhat.com/docs/manuals/linux>

#### 5. Changes to the network configuration

After installation we had to make changes to load the iucv driver. The installation did not do this automatically. Telnet is disabled after the install, so if you use Telnet you need make the change to allow Telnet to work.

#### 6. Oracle9i installation

The only difference with installing on SuSE is that when using the `cpio` command to extract the files, Red Hat did not support the "c" flag.

#### 7. Installing 9.0.2.0.2 patch set

When we installed the latest patch set, several files could not be copied. The two files `isqlplus` and `tnlsnr` in `$ORACLE_HOME/bin` could not be copied over until we renamed the old files. After renaming each file we clicked the **Retry** button and the installation continued.

The `genksms` command, which is missing in the 9.2.0.1 CD-ROMs, is included the 9.2.0.2 patchset. We were able to change the size of the SGA to 1 GB as documented in the Release Notes.

## 3.3 Downloading the code

The code can be downloaded from the Oracle OTN site, or you can use the CD-ROM media available at:

<http://www.store.oracle.com>

The steps to download the code are:

1. Go to: <http://www.otn.oracle.com>
2. Choose **Download** on the left.
3. Choose **Oracle9i Database**.
4. Choose **Oracle9i database for Linux/390 NEW**.
5. You may have to complete a series of questions about licensing to proceed.

Once the code is downloaded, go to the directory where you want to store the images:

1. `cd /oracle1`
2. `mkdir images`
3. `cd images`
4. FTP to the Linux guest:

```
lnx390_db_9201_Disk1.cpio.gz 440724391 bytes
```

```
lnx390_db_9201_Disk2.cpio.gz 616447271 bytes
```

5. `ls -la`

The `ls` command was to check that we had received the same number of bytes that we had downloaded.

6. Uncompress the files in the directory:

```
gunzip lnx390_db_9201_Disk1.cpio.gz
```

```
cpio -idcmv<lnx390_db_9201_Disk1.cpio
```

```
gunzip lnx390_db_9201_Disk2.cpio.gz
```

```
cpio -idcmv<lnx390_db_9201_Disk2.cpio
```

**Note:** Our version of SuSE Linux for S/390 came with FTP disabled. To enable, we completed the following steps:

1. Edit /etc/inetd.conf
    - a. Uncomment the line: ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd
  2. Stop and start inetd by:
    - a. Find the process number: `ps -ef | grep inetd`
    - b. Stop it: `kill -HU xxx`
    - c. Recheck to see if it is running: `ps -ef |grep inetd`
  3. The root ID is automatically disabled from being logged onto from FTP:
    - a. Modify /etc/users
    - b. Remove to rename root in the list.
1. Start inetd - `./inetd`

These are large files and the `ftp`, `gunzip`, and `cpio` commands can take 10 to 30 minutes to complete.

The size for the August 2002 download files were:

```
1nx390_db_9201_Disk1.cpio.gz 440724391 bytes
1nx390_db_9201_Disk2.cpio.gz 616447271 bytes
```

At this point, we were ready to run the Oracle Universal Installer.

### ***Web sites on OTN***

These are some other of the Web sites:

- ▶ To take a survey to provide feedback to Oracle:  
<http://www.oracle.com/go/?&Src=915044&Act=6>
- ▶ To check the Linux Forum:  
<http://www.otn.oracle.com/tech/linux/content.html>

At the bottom of this page there is the connection to the forum.

### ***Oracle discussion forums***

Oracle on Linux (this includes the Linux/390 entries)

FORUM: <http://www.oracle.com/forums/forum.jsp?id=374051>

### **Notes:**

1. For the downloads you will need to register for OTN.
2. Go to: <http://www.oracle.com> and choose **Oracle Technology Network** on the left hand side.

3. There have been some problems experienced in completing the download. Try it with both Netscape and Internet Explorer.
4. Oracle 9i for Linux on S/390 requires the new SuSE Linux Enterprise Server (SLES) 7 for S/390 and zSeries code (Linux 2.4 kernel). It will *not* work with the earlier SuSE Linux for S/390 level. For more information on SLES 7 go to: [http://www.suse.de/en/products/suse\\_business/sles/sles\\_s390/](http://www.suse.de/en/products/suse_business/sles/sles_s390/)

## 3.4 Finding the documentation

Documentation can be found through: <http://www.otn.oracle.com> at:

<http://www.otn.oracle.com/docs/products/oracle9i/content.html>

You can obtain the release notes for Oracle9i for Linux/390 from the Oracle docs Web site:

[http://docs.oracle.com/database\\_mp.html](http://docs.oracle.com/database_mp.html)

We used the following files of generic documentation for Oracle9i on Linux:

- ▶ *Oracle9i Installation Guide Release 2 (9.2.0.1.0)*, A96167-01
- ▶ *Oracle9i Administrator's Reference Release 2 (9.2.0.1.0)*, A97297\_01
- ▶ *Oracle9i Release Notes, Release 2 (9.2.0.1.0) for Linux/390 Systems (31-Bit), August 2002*, A97686\_02

These release notes contain important information not included in the Oracle9i Installation Guide Release 2 (9.2.0.1.0), and the Oracle9i Administrator's Reference Release 2 (9.2.0.1.0) for UNIX systems. This document contains the following topics:

- Third-party software notices
- System requirements
- Documentation
- Installation issues
- Product-related issues
- Post-installation issues

**Note:** Although Linux/390 is not mentioned in the titles of the *Oracle9i Installation Guide Release 2 (9.2.0.1.0)* for UNIX Systems, and the *Oracle9i Administrator's Reference Release 2 (9.2.0.1.0)* for UNIX systems, information specific to Linux Intel in these books also applies to Linux/390, except where noted in the release notes.

## 3.5 Checking the Linux kernel settings

In Chapter 2 of the *Oracle9i Installation Guide Release 2 (9.2.0.1.0)*, Part No. A96167-01, there is a step that asks you to check your kernel parameters.

First we checked that we had an xWindows environment that is required by the GUI of the Oracle Universal Installer by executing the command:

```
xclock
```

### ***Kernal parameter settings***

The kernel parameters settings that need to be checked are SSHSEM and SHMMAX.

The steps that allow you to check are:

1. Logon as root
2. **cd /proc/sys/kernel**
3. **catsem**

Our response was: 250 2560000 32 1024

4. **catshmmmax.**

Our response was: 33554432

5. **cat shmmmi**

Our response was: 4096

In our case, everything was larger than needed, so we made no changes.

**Note:** There is the additional information about kernel parameters in the release notes:

The following information supplements the information on kernel parameters in the Oracle9i Installation Guide Release 2 (9.2.0.1.0) for UNIX Systems:

SHMSEG: This kernel parameter is no longer supported in current kernel distributions.

SHMALL: Leave this kernel parameter set to the default value. This supersedes instructions in the installation guide to change the value.

SHMMAX: Set this kernel parameter to half the size of the physical random access memory (RAM) available on your system. Do not exceed a value of 2,147,483,647 bytes

## 3.6 Setting up the group and user ID for Oracle

First, create two groups DBA and OPER by issuing the GROUPADD command.

Then create a user ID called Oracle.

After you have completed the install process, you will have to set the following environment variables in your .profile to be able to continue:

- ▶ ORACLE\_HOME
- ▶ ORACLE\_SID
- ▶ PATH=\$PATH:\$ORACLE\_HOME/bin
- ▶ DISPLAY

In one case, we were working on a system that had a previously installed version of the Oracle database. We had to set these two variables to ensure we were pointing to the new files:

- ▶ ORACLE\_BASE
- ▶ LD\_LIBRARY\_PATH

## 3.7 Set up your system for xWindows

The Oracle Universal Installer (OUI) requires that you have xWindows interface to execute. To enable xWindows we installed Exceed from Hummingbird in one case, and VNC in another. We found that if you connected locally to the same LAN (or a fast LAN) to the zSeries, your response time while executing the OUI was acceptable.

In one case, when we were installing in California on a zSeries in Poughkeepsie, the response time of interfacing through xWindows, made it impossible to complete the install. In this case, we found several alternatives:

1. Install a VNC server on a Linux Intel box that is on the same local LAN as the zSeries. Use a VNC client connected to the VNC server, to connect to the Linux guest. This meant most of the network traffic was between the VNC server and the local zSeries.
2. Use the non-interactive install process that Oracle describes in Chapter 3 of the *Oracle9i Installation Guide*, A96167. We describe our experiences with this in 3.15, “Using the non-interactive install process” on page 61.
3. Use an AIX or Linux client that has xWindows installed

### Enable the Linux guest system to handle xWindows

To enable your Linux guest to export the xWindows display to your workstation, you must on the VM Linux guest system:

1. Set the DISPLAY variable by issuing the command:  
**export DISPLAY=hostname:0**  
where hostname is the name of your workstation or the IP address of your workstation. If you are using VNC, the hostname is the IP address of your VNC server.
2. Execute the command as user root on the vnc server:  
**xhost +**
3. Test by running the command:  
**xclock &**  
If this displays a window with a clock, you are ready to start the OUI.  
To remove the clock that is displayed, we issued the **kill** command to end the process.

Examples of how to do this are shown in Appendix C, “Examples of setting up the xWindows Interface” on page 197.

## 3.8 Getting the JDK

Download from:

<http://www6.software.ibm.com/dl/dk1x130/dk1x130-p>

In the early distribution of Oracle9i, the jdk code had to be downloaded as it was not included on the CD-ROM. Go to:

<http://www6.software.ibm.com/dl/dk1x130/dk1x130-p>

Choose the **Developer Kit** for S/390 Linux 31-bit for 1.3.1SR3.

Then issue the following **tar** command:

```
tar zxvf IBMJava2-SDK-131.s390.tgz
```

Remember the name of the directory where you place this code, as you will be asked for this in the installation process.

## 3.9 Installing Oracle9i binaries

After completing the download, you will have many files in the Disk1 and Disk2 directories. If you follow the instructions in the *Oracle9i Installation Guide Release 2 (9.2.0.1.0)*, Part No. A96167-01, you should have no problems first installing a database binaries, and then installing a database instance.

This section describes the choices we made as we went through the process of running the Oracle Universal Installer.

## 3.10 Checking the environment

Ensure the user ID `oracle` can write to the `ORACLE_HOME` directory.

We had to logon as root and issue the command `chmod 777 oracle1`, as the directory had been created by the user ID root.

Choose the names for `ORACLE_HOME` and `ORACLE_SID`. Our choices were:

```
ORACLE_HOME    /oracle1/dbserv  
ORACLE_SID    ORA1
```

Check that the `DISPLAY` variable is set.

Logon on as user ID `oracle`. If you are logged on as root, then `su oracle`. If you use the command, `su - oracle` you must reset the `DISPLAY` variable, because the dash (-) causes you to execute the `.profile` of the user ID `oracle`. We found it was easier to have two XTERM windows open. One was logged on as user ID root and the other as user ID `oracle`.

## 3.11 Running the Universal Installer

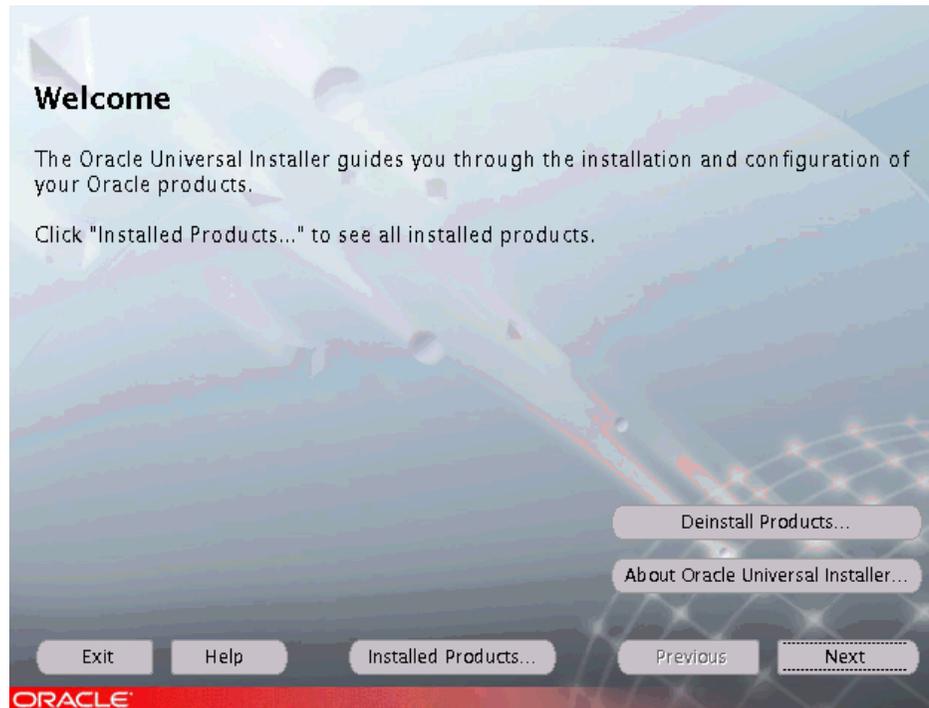
First go to the directory where the CD-ROM images were stored, then run the installer:

```
cd /oracle1/images/Disk1
./runInstaller
```

This will take a few minutes. Then a screen appears with the name of the Oracle Universal Installer, followed by the Welcome panel.

### ***Welcome panel***

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



## ***File Location panel***

At this point you are asked to indicate your inventory location directory.

### Inventory Location

**You are starting your first installation on this computer. As part of this first install, you will need to specify a base directory for installation files.**  
Another directory will be automatically set up for per-product installation files.  
The per-product installation files can vary in size, depending on the complexity of the products you install. You should plan on an average of about 15 0KB per product.

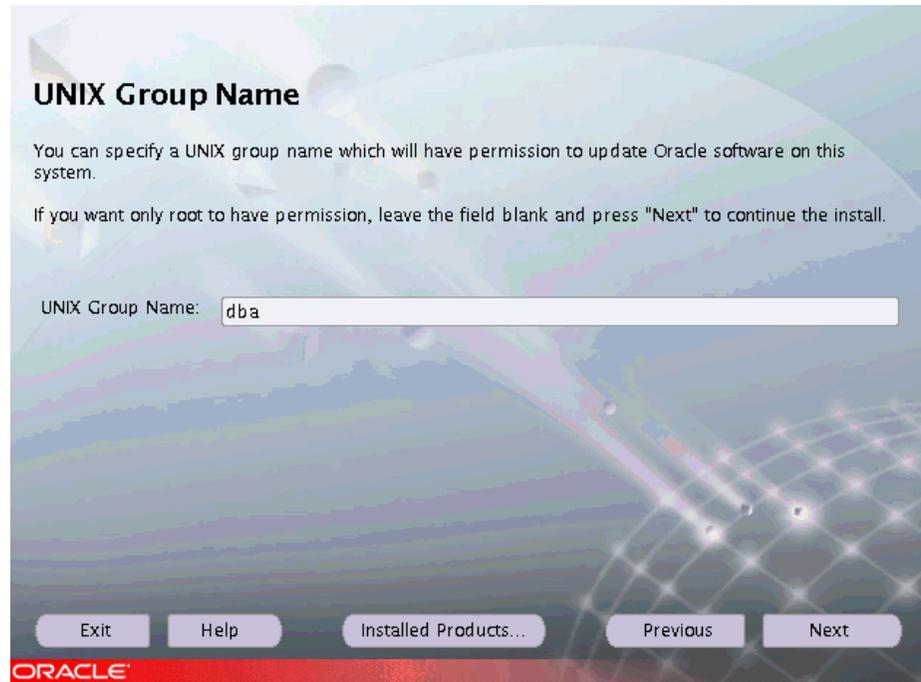
**The base directory:**  
\* Contains the permanent and per-product files in subdirectories.  
\* Must be writable by anyone installing or deinstalling products on this computer.

**What would you like as the base directory?**

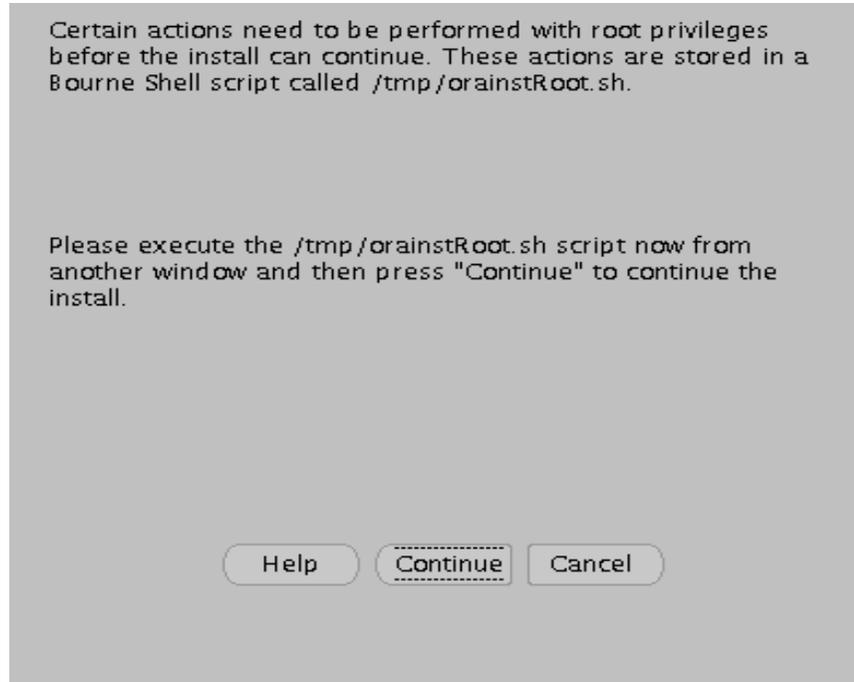
Inventory files will be in: /opt/oracle/orainventory  
Per-product files will be in: /opt/oracle/orainventory/Components

### ***UNIX group name***

Click to get the cursor at the beginning of the line. Enter dba (not in caps).



## Changing to root screen for orainstroot script

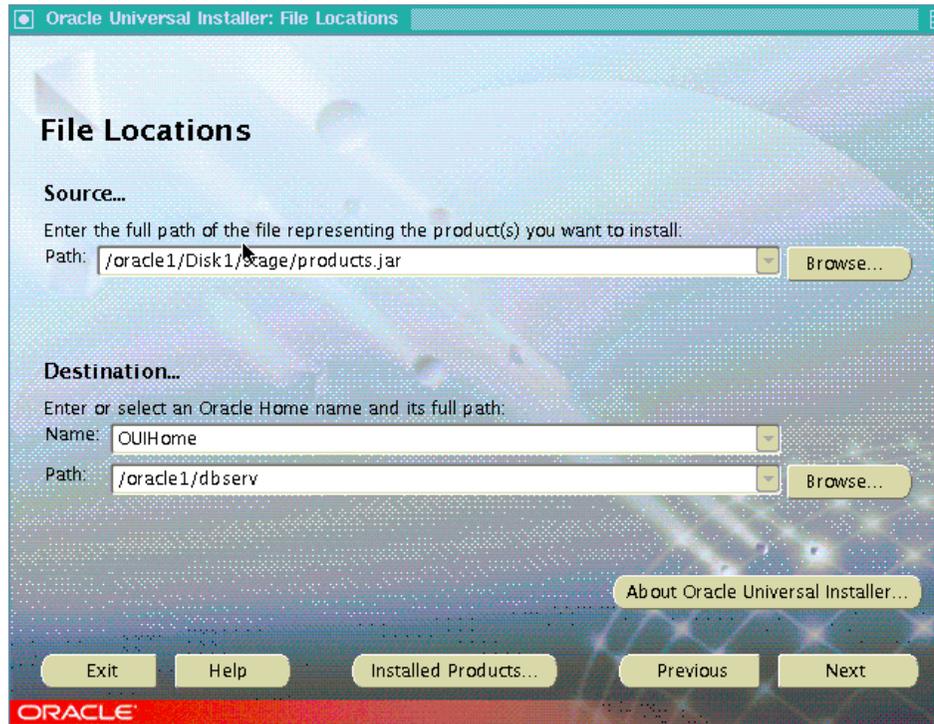


We returned to the Telnet session where we were logged on as root to execute the script:

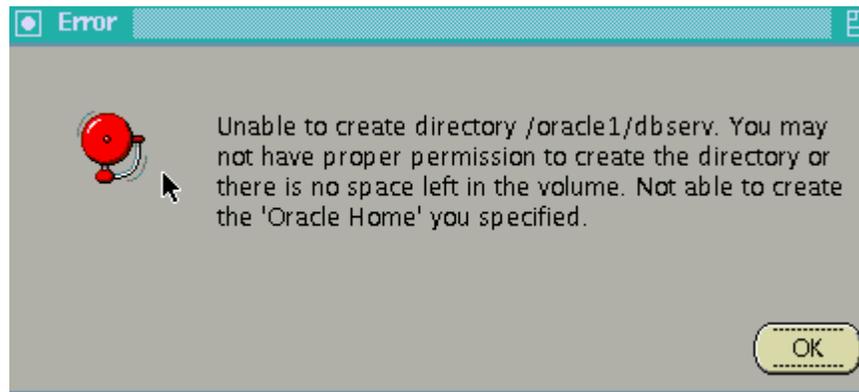
```
ioczm10s:/tmp # ./orainstRoot.sh
Creating Oracle Inventory pointer file (/var/opt/oracle/oraInst.loc)
Changing groupname of /opt/oracle/oraInventoryKA to DBA.
chgrp: invalid group name `DBA'
WARNING: chgrp of /opt/oracle/oraInventoryKA to DBA failed!
ioczm10s:/tmp #
ioczm10s:/tmp # ./orainstRoot.sh
Creating Oracle Inventory pointer file (/var/opt/oracle/oraInst.loc)
Changing groupname of /opt/oracle/oraInventory to dba.
ioczm10s:/tmp #
```

### File Locations

This is the screen that enables you to specify your ORACLE\_HOME. In our case, it was /oracle1/dbserv.

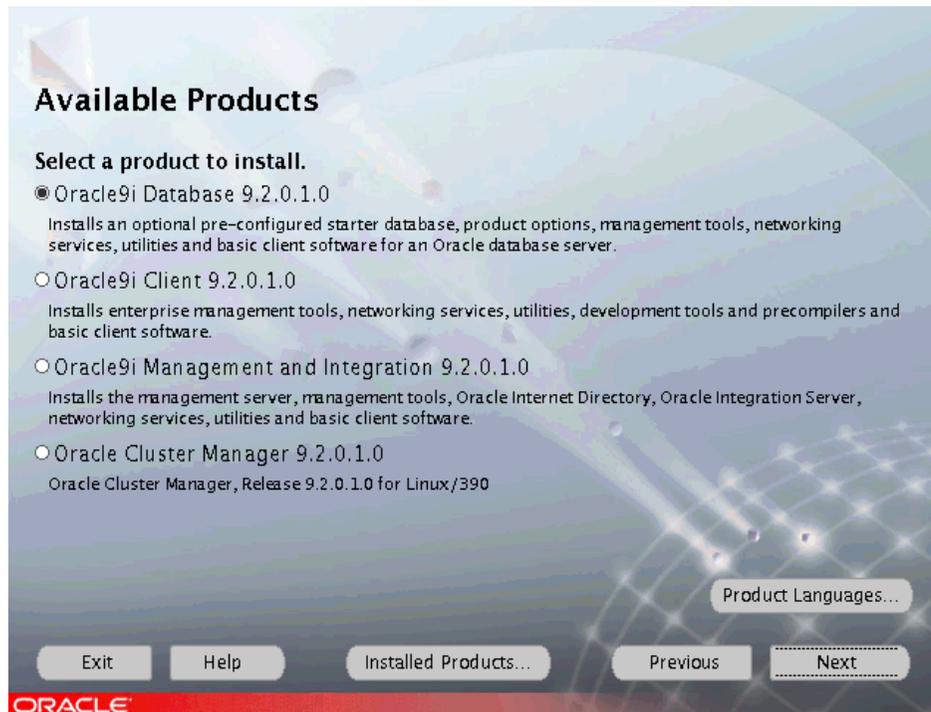


If you do not have write permission to the /oracle1 directory, you will get this message. You must resolve the issue before proceeding.



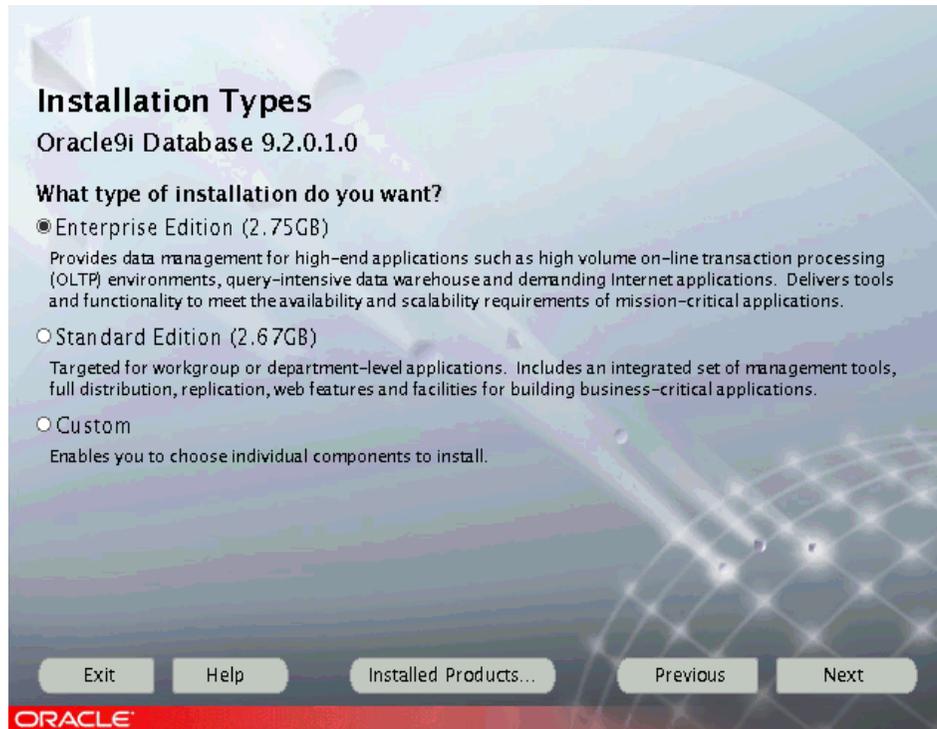
## Available products

There are four choices at this point. As we were installing the Oracle9i Database product, we choose the first option.



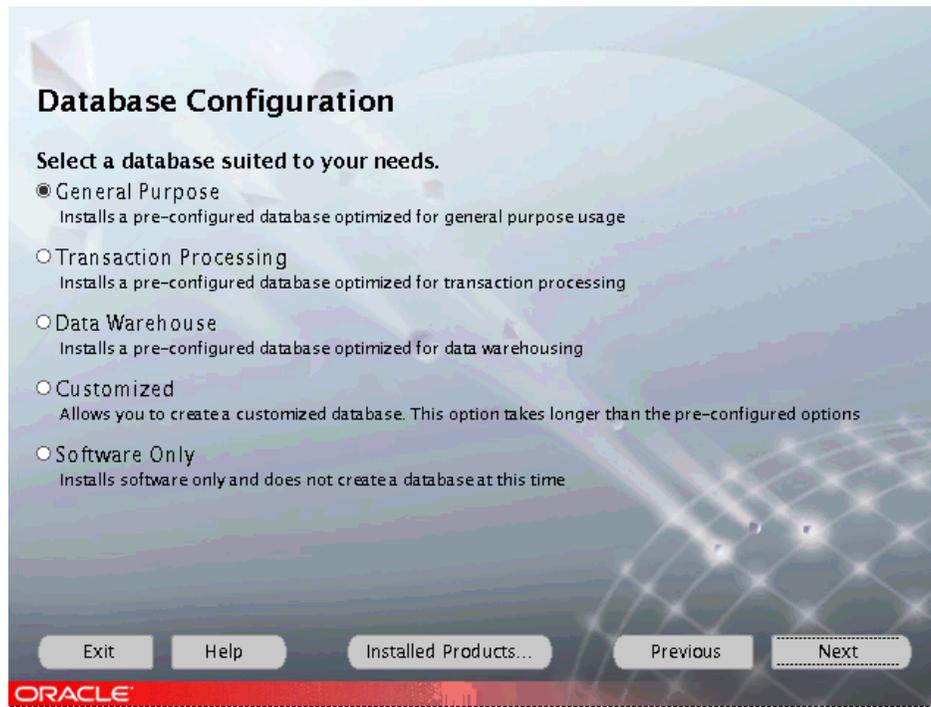
## ***Installation type***

We chose to install Enterprise Edition version of the database.



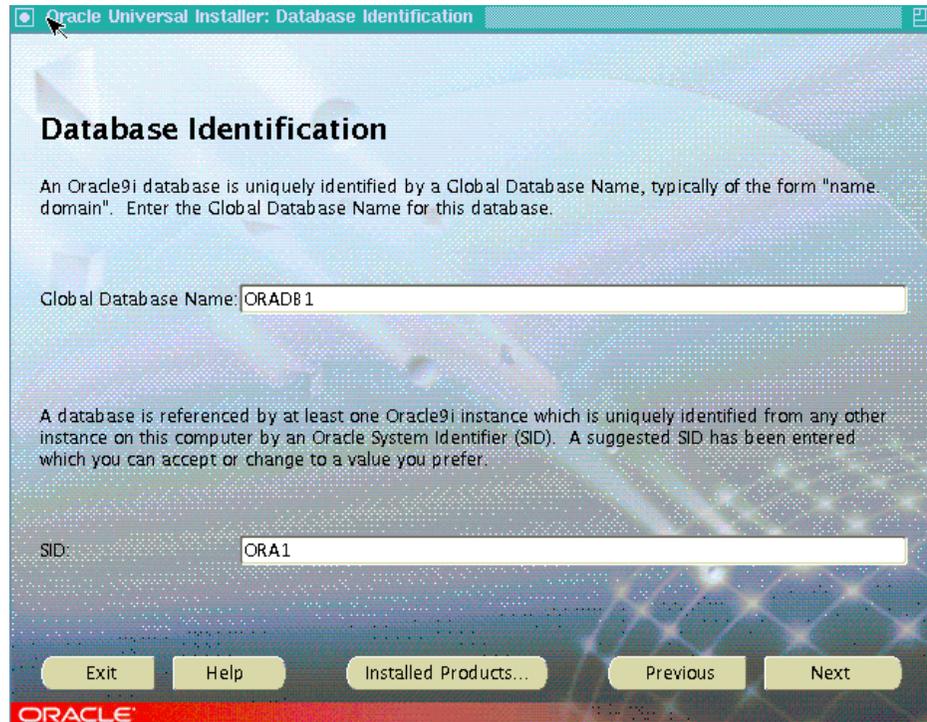
## ***Database configuration***

In our case, we choose to do a general purpose install that installs the binaries and creates a database.



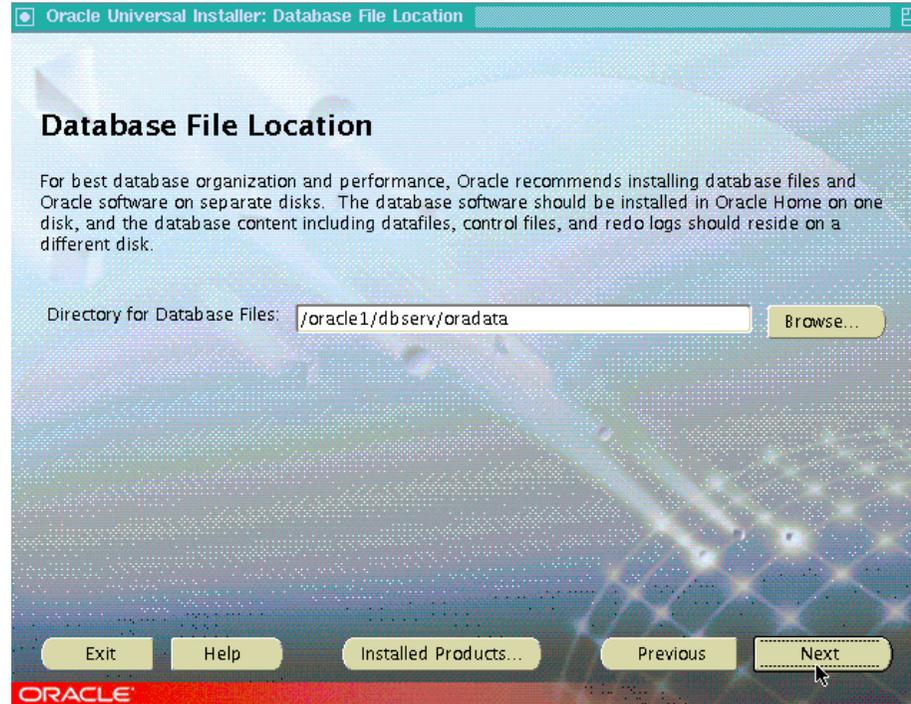
## Database Identification

We created the database ORA1 with the Global Database Name of ORADB1.



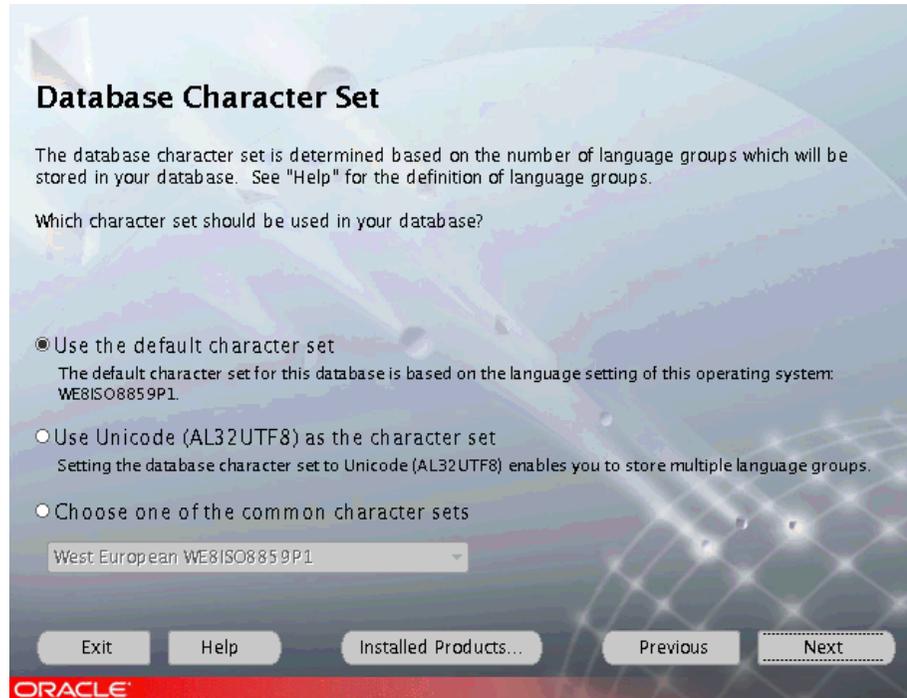
## **Database File Location**

At this point, you can choose to put your database file in a different directory if you want to.



## Database Character Set

We chose the default character set.



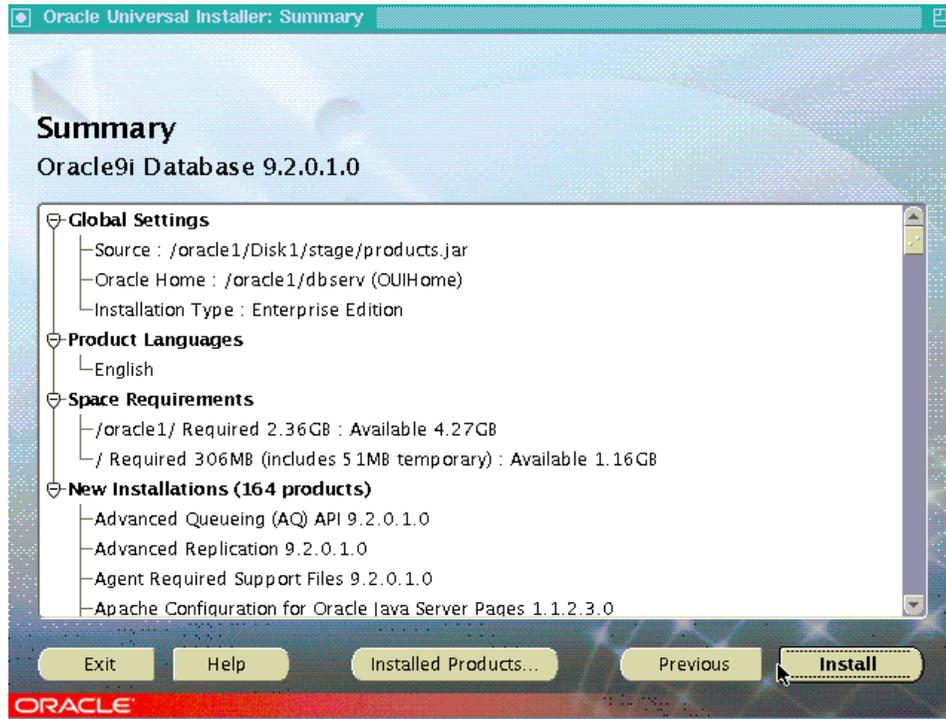
### **Location of JDK**

We had downloaded the JDK and installed in the /ora92 directory. The installer prompts you to browse to find the correct location.



## Summary

At the end of the series of questions, you will receive a summary of your choices to review.



Note the space requirements. If you do not have sufficient space it will prompt you to resolve this before proceeding.

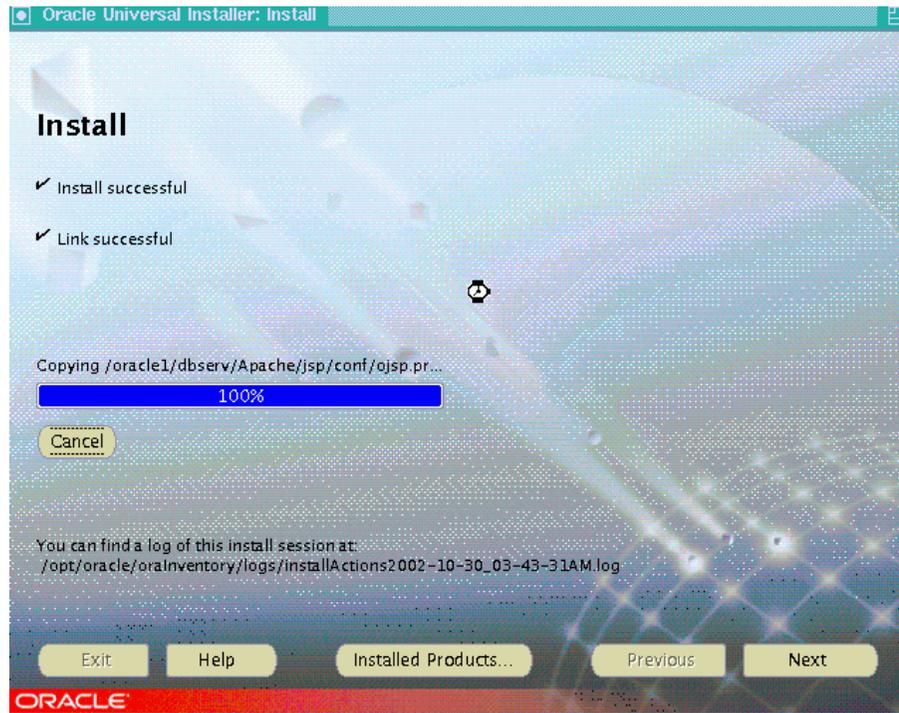
### ***Install completing***

This part of the installation takes a while as it is placing all the components of the database in the libraries.

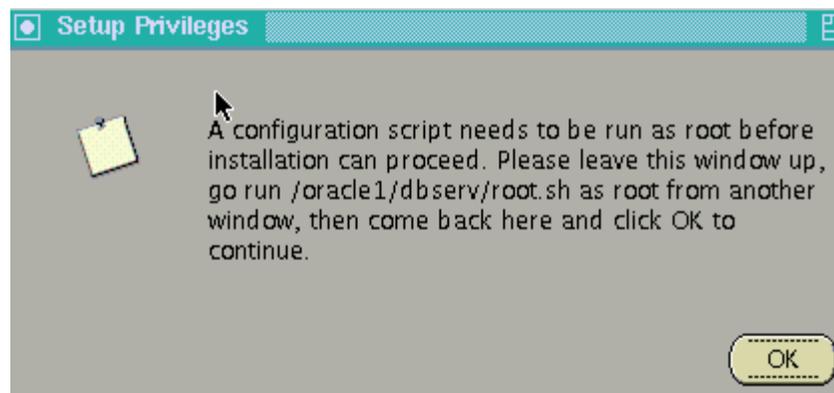


### ***Executing the root.sh script***

The installation process completes and then stops to ask you to log on as root and run the `root.sh` script.



This screen appears on top on the install screen.



Go to the XTERM screen that has the root user ID logged on and execute the **root.sh** script:

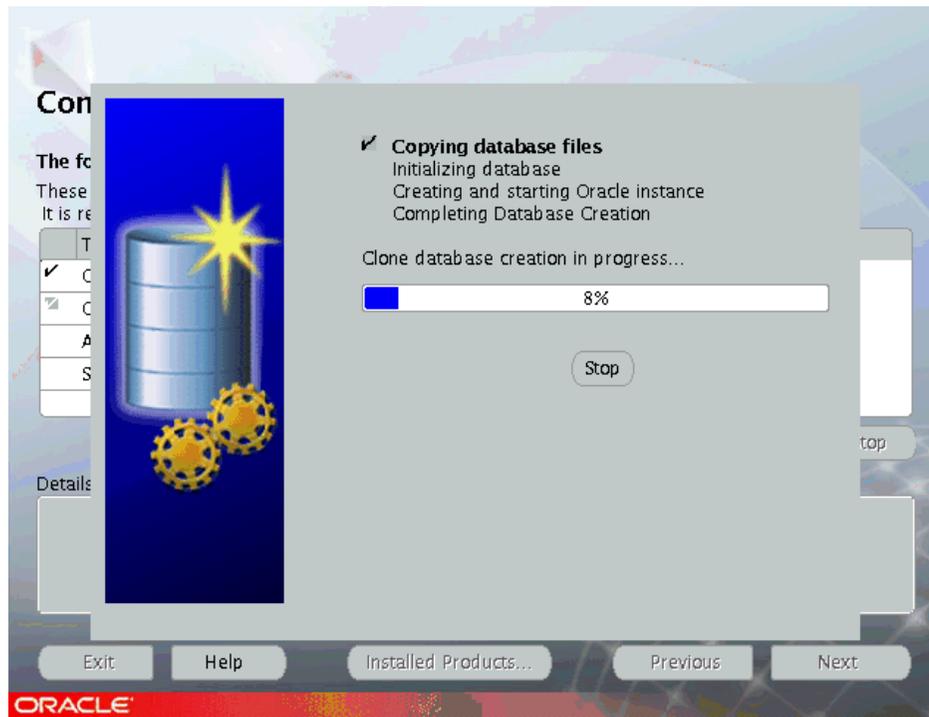
```
ioczm10s:/oracle1/dbserv # ./root.sh
Running Oracle9 root.sh script...
\nThe following environment variables are set as:
  ORACLE_OWNER= oracle
  ORACLE_HOME= /oracle1/dbserv

Enter the full pathname of the local bin directory: [/usr/local/bin]:
/usr/local
directory.../local/bin
  Copying ..ome to /usr/local/bin
  Copying ...env to /usr/local/bin
  Copying ...env to /usr/local/bin

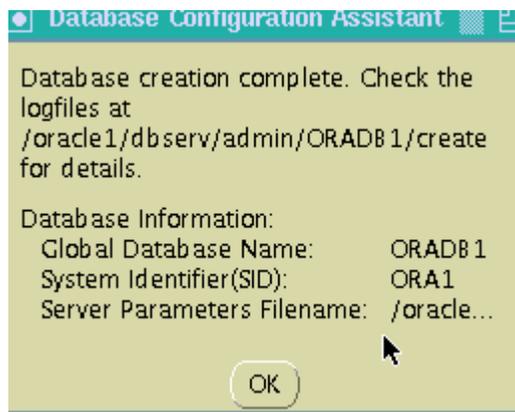
Adding entry to /etc/oratab file...
Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
ioczm10s:/oracle1/dbserv #
```

## Configuring the system

The installation process is now creating the database instance. This takes some time to complete.



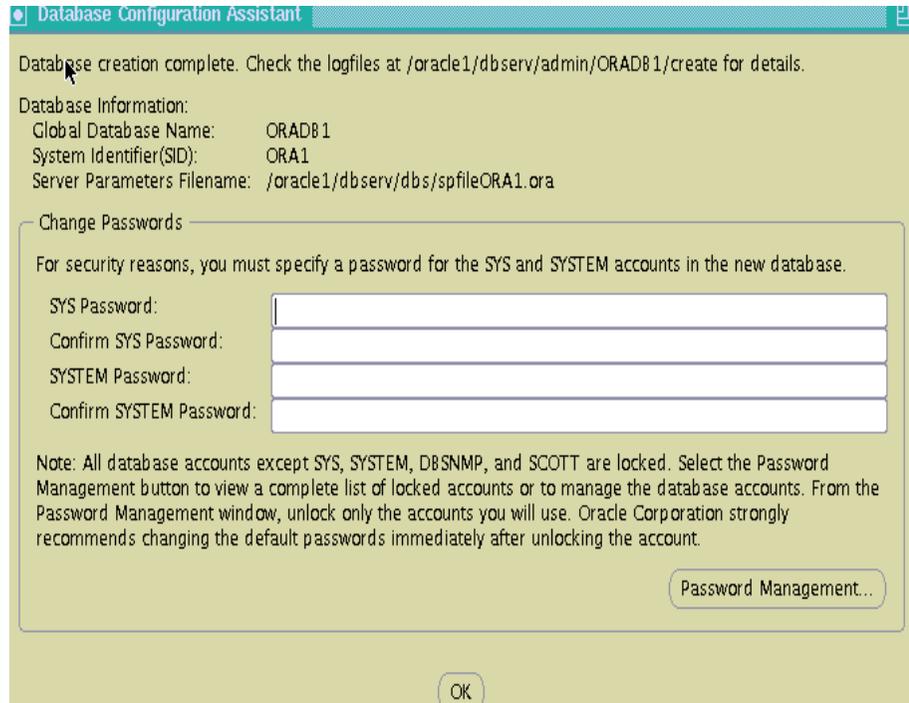
At the completion of the database creation, the following screen appeared.



This very small window needed to be resized so that the entire screen could be displayed, and so we could enter the passwords so the configuration process could continue.

### ***Choose the passwords***

You must choose the passwords for your database instance. We were not able to use the default options, such as manager for user ID system.



The screenshot shows the 'Database Configuration Assistant' window. The title bar reads 'Database Configuration Assistant'. The main content area has a light green background and contains the following text:

Database creation complete. Check the logfiles at `/oracle1/dbserv/admin/ORAD81/create` for details.

Database Information:  
Global Database Name: ORAD81  
System Identifier(SID): ORA1  
Server Parameters Filename: `/oracle1/dbserv/dbs/spfileORA1.ora`

**Change Passwords**

For security reasons, you must specify a password for the SYS and SYSTEM accounts in the new database.

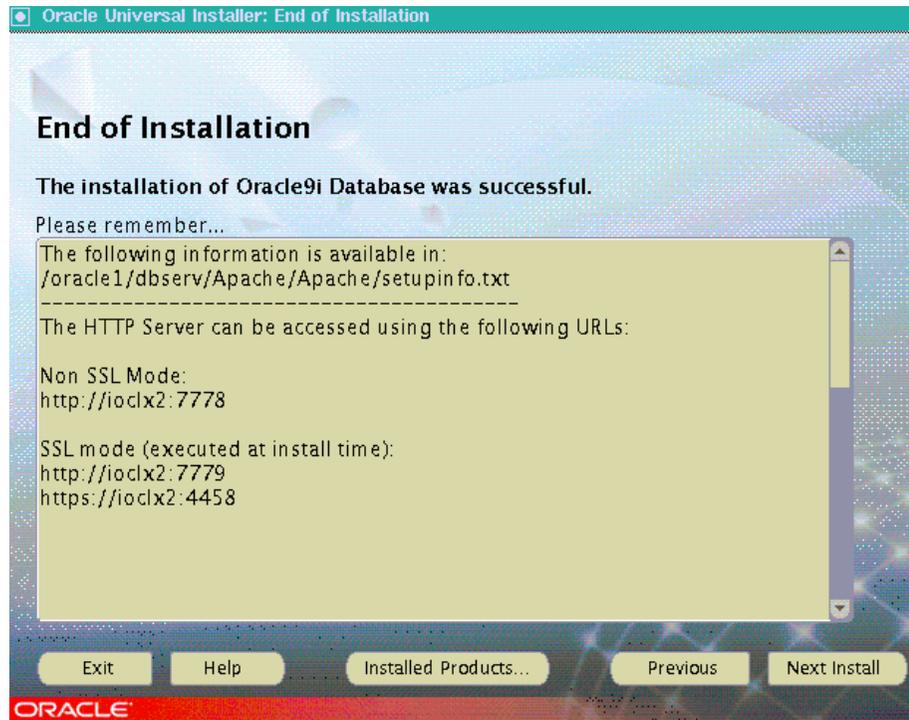
SYS Password:   
Confirm SYS Password:   
SYSTEM Password:   
Confirm SYSTEM Password:

Note: All database accounts except SYS, SYSTEM, DBSNMP, and SCOTT are locked. Select the Password Management button to view a complete list of locked accounts or to manage the database accounts. From the Password Management window, unlock only the accounts you will use. Oracle Corporation strongly recommends changing the default passwords immediately after unlocking the account.

Buttons: 'Password Management...' and 'OK'.

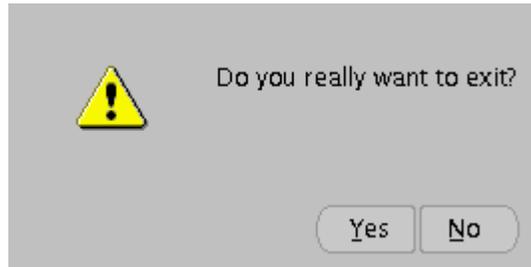
### **Successful installation**

At the completion of the process, the end of installation screen appeared.



## Exit

At this point, you have installed the Oracle binaries and created a database instance, so you can exit from the Oracle Universal Installer by clicking the **Yes** button.



In our case, we choose the one step process to install the Oracle libraries and to create the database instance.

You can choose to do this in two steps by choosing the **Software Only** option as shown in the screen in , “Database configuration” on page 46. This option will only install the libraries. You can then create the database by using the Database Configuration Assistant by executing the command `./dbca`. This will present a series of screens that will enable you to create the database instance.

## 3.12 Setting up the .profile

In `/etc` there is a profile that every user executes, but it is recommended not to use that for individual users. We created a `.profile` in home directory of the Oracle user ID, which in our case was `/opt/oracle`. We put the following entries in the Oracle `.profile`:

```
export ORACLE_HOME=/oracle1/dbserv
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_SID=ORA1
echo "your ORACLE_SID is ORA1"
```

## 3.13 Connecting with a local client

To test that the database was up, we set variables to allow us to connect to the database instance:

```
oracle@linux1:/oracle1/dbserv > export ORACLE_HOME=/oracle1/dbserv
oracle@linux1:/oracle1/dbserv > export ORACLE_BASE=$ORACLE_HOME
oracle@linux1:/oracle1/dbserv > export PATH=$PATH:$ORACLE_HOME/bin
```

```

oracle@linux1:/oracle1/dbserv > export ORACLE_SID=ORA1
oracle@linux1:/oracle1/dbserv > export
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib:/usr/local/lib:/usr/openwin
/lib

```

Then connected using sqlplus:

```

oracle@linux1:/oracle1/dbserv > sqlplus
SQL*Plus: Release 9.0.1.0.0 - Production on Fri Sep 6 18:09:58 2002
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.
Enter user-name: system
Enter password:
Connected to:
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production

SQL> select INSTANCE_NAME, HOST_NAME, VERSION, STARTUP_T, STATUS, * from
v$instance;

INSTANCE_NAME HOST_NAME  VERSION      STARTUP_T  DATABASE_STATUS
-----
ORA1           linux1      9.2.0.1.0    06-SEP-02  OPEN

SQL>

```

Now we were ready to clone this to create Linux guests and share the Oracle binaries.

## 3.14 Connecting a remote client to the database

To connect to a database from a remote machine you need to:

- ▶ Update the tnsnames.ora on your remote machine.
- ▶ Start the listener.
- ▶ Run **sqlplus**

Our entries are shown in 3.14.1, “Updating tnsnames.ora” on page 59 through 3.14.3, “Testing the connection” on page 60.

Also, you need to make sure you have connectivity between both machines.

### 3.14.1 Updating tnsnames.ora

```

=====
Client tnsnames.ora created manually
=====

```

```

#
# TNSNAMES.ORA
#
# Net8 definition for the database
#

ORA1 = (DESCRIPTION=
        (ADDRESS=(PROTOCOL=tcp)(HOST=ioclx)(PORT=1521))
        (CONNECT_DATA=(SID=ORA1))
      )

#

# Intermedia

#

extproc_connection_data =
  (DESCRIPTION=
    (ADDRESS_LIST =
      (ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROCVIS))
    )
    (CONNECT_DATA=
      (SID=PLSExtProc)
      (PRESENTATION = R0)
    )
  ) )

=====

```

### 3.14.2 Starting the listener

Log on to database server and set up your environment (ORACLE\_HOME, ORACLE\_SID, PATH):

```
oracle> lsnrctl start
```

### 3.14.3 Testing the connection

Run **sqlplus**

```
> sqlplus system/ibmoracle@ORA1
```

```
SQL*Plus: Release 8.1.6.0.0 - Production on Mon Sep 16 14:17:01 2002
```

```
(c) Copyright 1999 Oracle Corporation. All rights reserved.
```

```
Connected to:
```

```
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
```

SQL>

## 3.15 Using the non-interactive install process

In one case, we installed using the response file. The steps were:

1. Perform the same pre-installation steps as for an interactive install.
2. Use the user ID of Oracle with the home directory of /home/oracle, and:

```
ORACLE_HOME=/oracle
```

3. Install IBMJava2-s390-131 in /oracle.
4. Use a Linux text editor to create the oraInst.loc file. In our case, we put it in:

```
/var/opt/oracle
```

5. The oraInst.loc points you to the oraInventory directory for the install. It contains the following:

```
inventory_loc=/home/oracle/oraInventory
```

```
inst_group=dba
```

Both of these can be different on your machine.

6. Create a .profile in the /home/oracle directory with the line:

```
PATH=$PATH:/oracle/IBMJava2-s390-131/jre/lib
```

This allows the Installer to find the Java SDK.

7. Create a directory under /oracle for the response file:

```
mkdir response
```

Our response file was /oracle/response/ora92.rsp. This is shown in detail in Appendix F, "Sample of response file" on page 217. Modify the response file with your detailed information.

8. Now you are ready to start the response file install, on the Shell Konsole enter the following at the prompt:

```
oracle> /oracle/Disk1/runInstaller -silent -responsefile  
/oracle/response/ora92.rsp
```

9. The following was displayed on the Linux KDE shell, which used the Konsole:

```
Initializing Java Virtual Machine from  
../../stage/Component/oracle.swd.jre/1.3  
.1.0.0/1/DataFiles/Expanded/lnx390/IBMJava2-s390-131/jre/bin/java. Please  
wait ...
```

```
oracle > In Unix Group Name Page
In Cluster Node Selection Page
In File Location Page
In Available Products Page
In Installation Type Page
In Component Location Page
In Title Page
In Title Page
In Choose Directory Pasge
In Title Page
In Choose Directory Page
In Summary Page
In End of Installation Page
The installation of Oracle9i was successful.
Please check
/home/oracle/oraInventory/logs/silentInstall2002-10-14_12-04-29PM.log for
more details
```

10. DBCA displays password change screen for sys and system user IDs. Change the passwords and continue.
11. Check the content of the silentInstall.log. This was our log:  

```
SilentUnstall2002-10-14_12-04-29PM.log
```

A configuration script needs to be run as root before install can proceed. Please leave this window up, go run /oracle/OraHome1/root.sh, then come back here and click **OK** to continue \*\*\*

The installation of Oracle9i Database was successful.
12. Log on as root and run the **root.sh** script.
13. Review the InstallActions log in the same directory.
14. Update your .profile in /home/oracle as shown in 3.6, "Setting up the group and user ID for Oracle" on page 36, and 3.12, "Setting up the .profile" on page 58.

This is the completion of the install using the response file.

## 3.16 Restarting the OUI

There are several files that Oracle uses to keep track of the choices you make while using the Oracle Installer. If you have to restart the process, there may already be entries in these files that can direct the OUI to default values. The best way to restart is to use the OUI to de-install and then restart. However, if you have to manually restart, you should look at these files and remove the entries or make the desired changes:

- ▶ `/var/opt/oracle/oralnst.loc` contains these entries:
 

```
inventory_loc=/oracle1/dbserv
inst_group=dba
```
- ▶ `/opt/oracle/.profile` contains:
 

```
export ORACLE_HOME=/oracle1/dbserv
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_SID=ORA1
echo "your ORACLE_SID is ORA1"
```

You may want to change these values or remove the entries.
- ▶ `/etc/oratab`  
This file has entries for the `ORACLE_HOME` and `ORACLE_SID`
- ▶ `$ORACLE_HOME/dbserv/` has several files such as `inventory_loc`.

Remember, the best way to restart is to direct the OUI to use the de-install option.

## 3.17 Testing the de-install process

In one case, we decided to start over from the beginning, so we used the de-install option to clean our database (that had a SID of kao1) and files.

Make sure the database is shutdown before starting this process.

We started the Installer by issuing the command `./runInstaller`, then we clicked the **De-install** button. The following screen appeared.

### Installing a patch using the `opatch` command

Oracle patches are applied using the *Opatch* utility that is supplied with the patch itself. However, early copies of this utility did not recognize the Linux for zSeries platform, so it failed (opatch version 1.0.0.13 and earlier).

The *operational solution* is to edit a configuration file supplied.

With the patch and change the platform number to the linux-intel platform number. The “readme” supplied with the patch instructs how to install it, for example:

```
#"Patch Installation Instructions:
# -----
```

```
# To apply the patch, unzip the PSE container file:
#
# % unzip 2648655.zip
#
# Set current path to the directory containing the patch:
#
# % cd 2648655
#
```

At this point issue the command:

```
cd etc/config
```

Edit the file called inventory.

Change the line that reads: <platform name="Linux for S/390" id="211"/>

To <platform name="Linux for S/390" id="46"/>

The patch can now be applied successfully, however, a further error will occur when Opatch attempts to document what it has done. This is due to a wrongly named directory; it can be corrected by creating a symbolic link:

```
cd $ORACLE_HOME
```

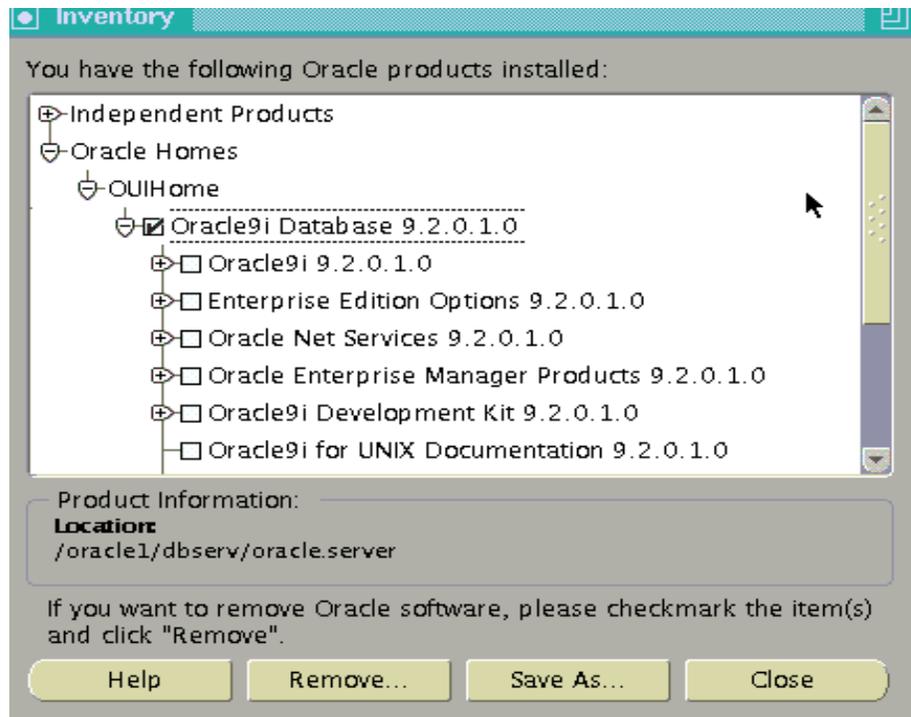
```
cd ../oui/bin
```

```
ln -s linux Linux
```

We also had to create a symbolic link in the etc directory for oralnst.loc with link to oralnst.loc in the /var/opt/oracle directory.

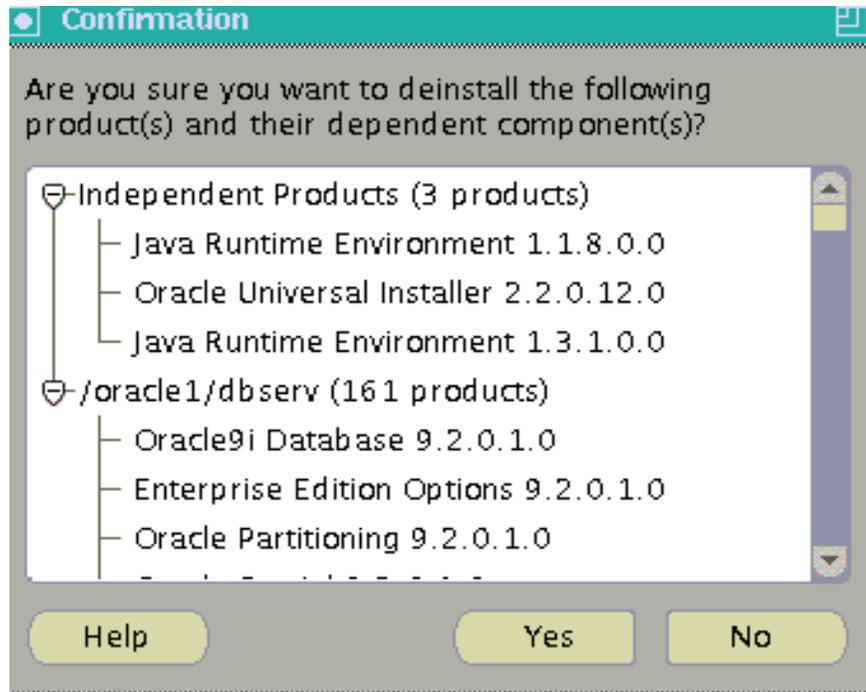
Then **cd** back to the patch directory proceed as documented in the readme.

## List of Oracle products to remove



OUIHOME was the Oracle Home that we wanted to delete, so we checked this entry. The next screen asked that we confirm the components we wanted to delete.

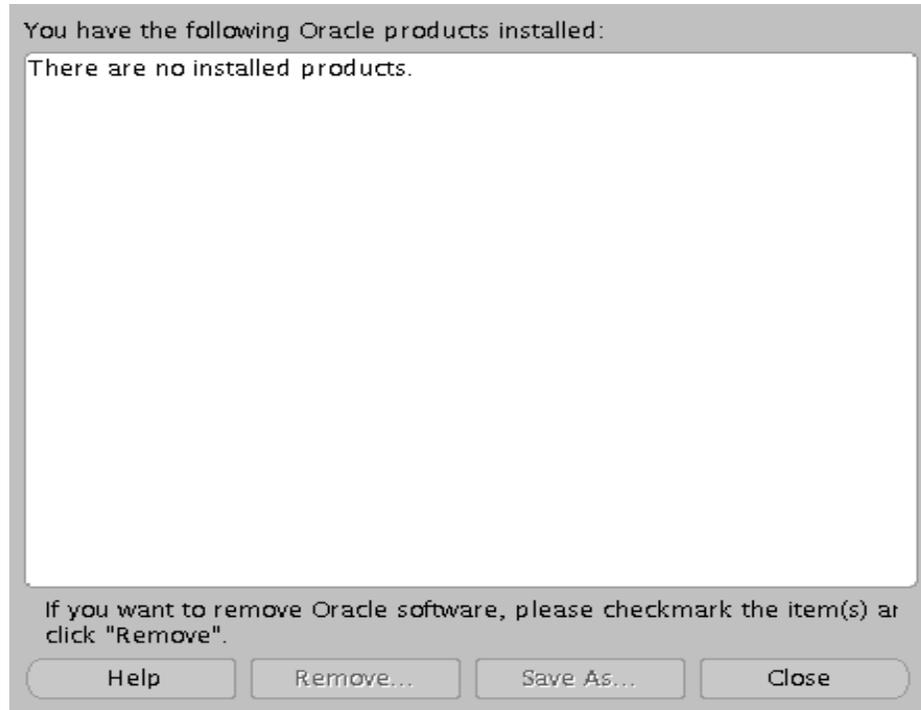
## List of products and components to de-install



After the de-installer removed these products and components, the database instance was no longer there and the database files were removed.

After the de-install process was completed, we tried a second time to start the de-installer. The following screen showed that we had nothing installed to remove.

## No products left to remove



However, we found that all the Oracle binaries were still present. So, to start a complete reinstall, we went to the `ORACLE_HOME` directory and issued the `rm -r *` command to remove all files in the directory.

We also removed the files in `/var/opt/oracle` as noted in 3.16, "Restarting the OUI" on page 62.

At this point, we were ready to restart the installation process.





## Creating Linux virtual machines with shared Oracle binaries

This chapter describes when you should consider sharing Oracle binaries, and the steps we executed to create multiple virtual Linux images for our Oracle testing.

## 4.1 Overview

Over the years, one of the key benefits of using VM has been the ability to share mini disks in a read only mode for CMS users. There was one copy of the data that was always current, saving disk space, and making only one copy to update.

With the new offering of Oracle9i for zLinux, we explored the possibility of having one copy of the Oracle libraries (binaries) shared by several Linux guests and thus several database instances, to see if we could save disk space and reduce maintenance efforts. Sharing the binaries with a shared minidisk is similar to the concept of sharing files on an NFS mounted disk.

We came to the conclusion that there are some situations where you might want to take this approach, and other situations where you might not. This chapter describes the pros and cons of sharing Oracle binaries, and then gives an example of how we set up our test environment.

### ***Pros and cons of using shared Oracle Binaries under z/VM***

We setup one minidisk, accessible to all the Linux guests, which contained all the Oracle files that are used in a read only mode. Each Linux guest had a second disk where all the modifiable files were stored such as logs and data files. We only tested basic queries. We did not test all possible situations such as using Java against the database, stored procedures written in either Java or PL/SQL, or the Oracle's OCI interface.

We tested database creation, table creation, insert row, delete row, select statements, drop table, drop database functions, and we came to the following conclusions:

- ▶ Only one Linux guest can be given read/write access to the shared disk. All other must have read only access to this disk. When the Linux guest with the read/write access is up and running, all other Linux guests must not be active. When the disk is being shared by several Oracle database instances, there can be no updates by Linux system that has read/write access.

In our case LINUX1 had read write access. LINUX2, LINUX3, LINUX4 had read only access. So when LINUX2, LINUX3, and LINUX4 were running with their database instances sharing the binaries, LINUX1 did not have read write activity on the shared disk. When LINUX1, needs to do updates to the Oracle shared binaries, the other three LINUX guests must not be running,

This approach of easily cloning and sharing the Oracle binaries, could be very productive in an educational environment, a test or development environment, or a smaller installation where one DBA managed several instances.

One very large customer raised several operational issues with sharing the Oracle binaries.

Care must be taken when applying patches or migrating to a new release.

In one case where there were many DBAs handling many databases on several platforms, this approach could not be used for their production databases as it would not conform to their standard procedures. One concern is that an inexperienced system or database administrator can mistakenly take down all Oracle instances sharing the single set of binaries. The impact of outages that can be caused by sharing binaries should be considered prior to making a commitment to sharing binaries in a critical production environment.

- ▶ In a RAC environment, the Oracle Universal Installer expects to install binaries on all nodes, so it is not recommended in a RAC environment using raw devices. Oracle is developing a cluster file system that should allow sharing, so this might change in the future.
- ▶ You must be very careful to only put the read only files on the shared disk. Oracle does not provide a definitive list of these files, and they may change from release to release. Since we did not check all possibilities, we may not have identified all the sharable files.
- ▶ As this is an operational procedure and currently not supported by Oracle at this point in time, any tar that you open with Oracle should be about problems with the database in a configuration that is not sharing binaries. So you might want to insure you have a process in place to move an Oracle database to its own set of binaries should you have to work with Oracle support to resolve a problem. This would require re-testing the problem to ensure that it was not caused by the disk sharing configuration. This may be acceptable in some environments, such as the provisioning of training classes, and unacceptable in other situations, such as business applications running in production.

If you are willing to test your specific situation, we feel there are situations where you could take advantage of this approach.

Certainly the ease of cloning Linux guests with an Oracle databases in this environment is an advantage you get when using this platform. Sharing binaries is something that will have to be evaluated for applicability in your situation.

The rest of this chapter describes our experiences in the areas of using cloning to produce multiple Linux guests with Oracle databases, and to set up using one copy of the Oracle binaries to be used by several Linux guests.

## 4.2 Overview of the tests

We will cover the following topics:

- ▶ Description of test
- ▶ Set up of the hardware:
  - IOCDs
  - DISK
- ▶ Set up of the operating system software:
  - VM
    - Directory setup for Linux virtual machines
    - Profile
    - Mini disk for sharing
  - TCPIP
    - PROFILE TCPIP
    - DTCPARMS
    - TCPIP DATA
- ▶ Set up of Linux
  - Install
  - BOOT from CD or FTP site or copy files to Local
  - Run YAST
  - Clone
- ▶ Installing Oracle
  - Install on share disk
- ▶ Conclusion

In the mainframe world, disks are called DASD, however, in this document we will refer to DASD by using the term *disk*.

### 4.2.1 Description of test

The purpose of the test was to verify that we could share Oracle binaries across multiple Linux virtual machines under virtual machine (VM) and create several databases using the one set of binaries.

We tested five Linux virtual machines under VM sharing one set of Oracle binaries.

Each Linux virtual machine had its Linux binaries installed on a dedicated disk, and had access to shared a VM minidisk where the Oracle binaries were installed. On each Linux machine we created a database with control files, database files, and log files stored locally.

## 4.2.2 Hardware setup

The hardware we used was a Mutiprise 3000 with 2 LPARs; one LPAR was running MVS, and the other LPAR was running VM. In this document we will focus on the LPAR running VM, as shown in the following diagram.

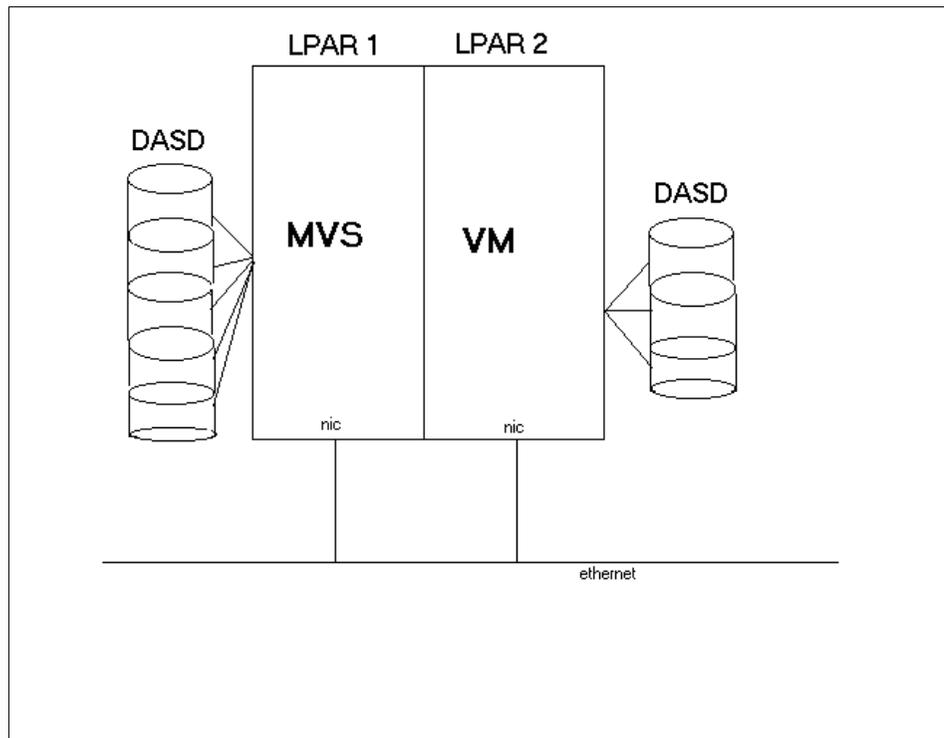


Figure 4-1 System with two LPAR definition

To have multiple LPARs, we needed to make changes to the IOCDs.

### IOCDs

The IOCDs is where you define your resources for the LPAR. Some of the resources defined are network cards, disk, tape drives, console, and any other devices you want to access on the LPAR.

The machine used was previously set up with two LPARs. The LPAR we used had a network card and enough disks allocated to complete the test.

### **Disk**

We needed to allocate disk for the OS. We had the option of having all the disks managed by VM, or some managed by VM, and others dedicated to Linux.

We allocated five disks to VM and one dedicated DISK for each Linux machine. Disks need to be initialized before they can be recognized by VM.

## **4.2.3 Set up of software**

### **VM**

VM is an operating system that can host many operating systems; these operating systems are hosted as virtual machines. To add virtual machines you need to make changes to the VM directory.

VM should be operational and access to the ID MAINT is required to do the setup for the Linux machines.

The following diagram shows the VM LPAR with the disk we allocated for each of the five Linux virtual machines.

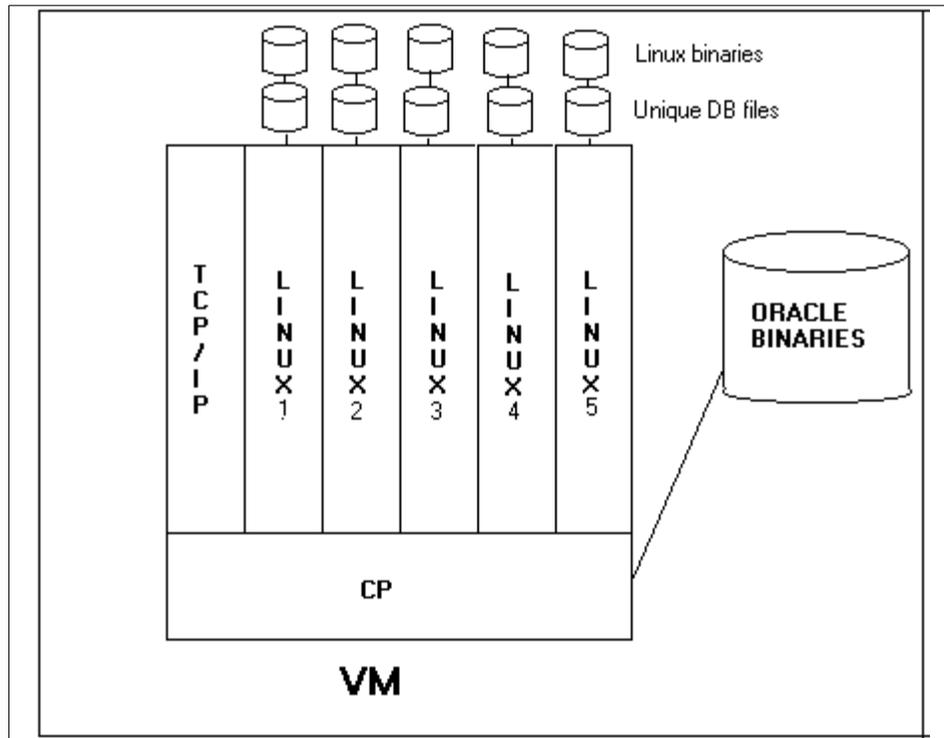


Figure 4-2 VM LPAR with 5 Linux guest

## Directory setup for Linux virtual machines

The user directory describes to CP (Control Program) the configuration and operating characteristics of each virtual machine that can be created by VM.

To create virtual machines for Linux, the user directory needs to be edited and an entry added for each virtual machine; the file that you edit is the USER DIRECT file.

Refer to *z/VM V4R3.0 CP Planning and Administration*, SC24-6043 for more information.

The USER DIRECT file is in the search path for the MAINT id and it can be located by doing the following command:

```
LIST USER DIRECT *
```

The following are entries for virtual machines LINUX1 and LINUX2; these entries were added to the USER DIRECT file. For virtual machines LINUX3, LINUX4, and LINUX5, the entries were similar to the entry for LINUX2:

```

USER LINUX1 LINUX1 480M 1000M BG
ACCOUNT 55 P390
MACHINE ESA
IUCV ALLOW
IUCV ANY
IPL CMS
CONSOLE 009 3215
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
LINK MAINT 190 190 RR
LINK MAINT 19E 19E RR
LINK MAINT 19D 19D RR
MDISK 191 3390 1194 010 240W02 MR ALL
MDISK 345 3390 0000 3338 LINUX1 MWV LWORK
DEDICATE 0306 0306

```

\*

```

USER LINUX2 LINUX2 480M 1000M BG
ACCOUNT 61 P390
MACHINE ESA
IUCV ALLOW
IUCV ANY
IPL CMS
CONSOLE 009 3215
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
LINK MAINT 190 190 RR
LINK MAINT 19E 19E RR
LINK MAINT 19D 19D RR
LINK LINUX1 345 345 RR
MDISK 191 3390 1194 010 240W02 MR ALL
DEDICATE 312 312

```

## Sharing binaries

To support the sharing of Oracle binaries across the Linux virtual machines, we needed to set up a shared disk. In VM you share a disk by creating minidisk, the minidisks are defined in the USER DIRECT file.

Please note the *highlighted* line in the above example.

The entries for the other Linux machines were all similar to the entry for virtual machine LINUX2; the entry for virtual machine LINUX1 is different because machine LINUX1 owned the disk where the binaries for Oracle were installed.

Virtual machine LINUX1 has the definition of the mini disk 345:

```
MDISK 345 3390 0000 3338 LINUX1 MWV LWORK
```

Virtual machine LINUX2 has the LINK entry in order to access mini disk 345:

```
LINK LINUX1 345 345 RR
```

This RR link was added to all the virtual machines.

**Note:** Our completed environment was one guest with read write (RW) access to the shared disk, and four guests with read only (RO) access to the shared disk.

The one guest that has RW access can only be up when the other guests have the disk unmounted, or are not active.

If you have to make the changes to the shared disk, stop the four guests with read only access, and then mount the disk on the guest with RW access. Make the changes, then shut it down, or unmount the disk.

Start the four guests and mount the shared disk as RO.

## Profile

For each virtual machine (user ID) the PROFILE EXEC file is executed each time you log on; this file needs to be updated with the line "ACCESS 345 L" to have access to the shared minidisk:

```
INDICATE
CP SET MSG ON
CP SET EMSG ON
CP TERM MORE 1 1
SET IMPCP ON
SET RUN ON
CP TERM CHARDEL OFF
CP SET PF01 RETRIEVE
CP SET PF12 RETRIEVE
CP SET PF12 RETRIEVE
CP SET RETRIEVE MAX
SET CASE M I
ACCESS 345 L
VMLINK TCPMAINT 592
```

## TCP/IP Setup under VM

TCPIP needs to be set up in order to have communication with the Linux machines. The files to edit for TCPIP are PROFILE TCPIP, SYSTEMS DTCPARMS, and TCPIP DATA.

## **PROFILE TCP/IP**

This is the file that contains the definition for the TCPIP network; in this file we define our private network and the method to be used to connect to the external networks.

We used IUCV protocol to communicate to the Linux virtual machines; the network card is allocated to VM. The Linux machines are on a private network, using the VM as a gateway to the outside world.

The private network used was 10.0.0.0 with the following IP address:

```
LINUX1  10.0.0.2
LINUX2  10.0.0.3
LINUX3  10.0.0.4
LINUX4  10.0.0.5
LINUX5  10.0.0.6
```

IP address 10.0.0.1 is used as the gateway to the outside world.

IP address of 9.38.159.11 is used for IOCZM10S.

Below is the definitions added to PROFILE TCPIP file:

```
DEVICE LCS1 LCS E42
LINK IOCZM10S ETHERNET 2 LCS1
DEVICE LINUX1 IUCV 0 0 LINUX1 A
LINK LINUX1 IUCV 0 LINUX1
DEVICE LINUX2 IUCV 0 0 LINUX2 A
LINK LINUX2 IUCV 0 LINUX2
DEVICE LINUX3 IUCV 0 0 LINUX3 A
LINK LINUX3 IUCV 0 LINUX3
DEVICE LINUX4 IUCV 0 0 LINUX4 A
LINK LINUX4 IUCV 0 LINUX4
DEVICE LINUX5 IUCV 0 0 LINUX5 A
LINK LINUX5 IUCV 0 LINUX5
; -----
; Define the internet (IP) address(es) for this VM host
; -----
HOME
9.38.159.11 IOCZM10S
10.0.0.1 LINUX1
10.0.0.1 LINUX2
10.0.0.1 LINUX3
10.0.0.1 LINUX4
10.0.0.1 LINUX5
; (End of HOME address information)

GATEWAY
```

```

; (IP) Network First      Link      Max. Packet Subnet      Subnet
; Address      Hop        Name      Size (MTU)  Mask       Value
; -----
9              =          IOCZM10S  1492       255.255.255.255 9
10.0.0.2      =          LINUX1    1492       HOST
10.0.0.2      =          LINUX1    1492       HOST
10.0.0.3      =          LINUX2    1492       HOST
10.0.0.4      =          LINUX3    1492       HOST
10.0.0.5      =          LINUX4    1492       HOST
10.0.0.6      =          LINUX5    1492       HOST
;
; Define The DEFAULT route used for any network not explicitly routed
; via the previous entries.
; -----
;
DEFAULTNET 9.38.159.1 IOCZM10S 1492 0
;
; (End of GATEWAY Static Routing information)

; -----
; Start all network interface devices used in this environment.
; -----
START LCS1
START LINUX1
START LINUX2
START LINUX3
START LINUX4
START LINUX5

```

### **SYSTEM DTCPARMS**

In this file we need the definition of the TCPIP servers. In our case, we only added the two lines for the tcpip server definition to the file SYSTEM DTCPARMS:

```

:nick.tcpip :Type.server :Class.stack
           :Diskwarn.90

```

### **TCPIP DATA**

In this file we have data to be used by the TCPIP clients.

The only modifications to the system were the following lines:

```

HOSTNAME IOCZM10S
DOMAINORIGIN SANMATEO.IBM.COM
NSINTERADDR 9.38.156.250

```

The rest of the TCPIP DATA file was left as it was.

This completed the initial setup. We are now ready to install Linux.

## 4.2.4 Set up Linux

There are several documents outlining the Linux installation for zSeries or S/390. The most concise document is the Redpaper *Building Linux Systems under IBM VM*, REDP0120. All the documents have good information, and they were reviewed before the install was done for our test.

Make sure all necessary devices, including the appropriate console, DISK, and communications devices, have already been defined to the VM system.

Also, make sure you have access to the Linux distribution from VM.

The distribution of Linux we used was SLES-7 Kernel 2.4.7 - SuSE SMP on Linux/390. All the steps in this document relate to installing this distribution of Linux.

The first thing to do is boot Linux running in RAMDISK. There are several ways to do this: Boot from CD or FTP site, or copy files to VM disk, and boot from VM disk. During our testing we booted from CD, and also copied the files to a VM disk, and booted from the VM disk. Booting from the VM disk was the preferred method.

### Steps to get Linux running in RAMDISK

1. First use the following commands to copy three files from the SuSE CD, or distribution to the local VM disk:

```
/suse/images/vmrdr.ikr to SUSE IMAGE (this is the Linux kernel image).  
/suse/images/initrd to SUSE INITRD  
/suse/images/parmfile to SUSE PARM
```

2. If you choose to use FTP to copy these files, follow these instructions after you have connected to the site that has the distribution. It is assumed that the distribution is at the directory you connected to:

```
ftp>bin  
ftp>locsite fix 80  
ftp>get /suse/images/vmrdr.ikr suse.image  
ftp> get /suse/images/initrd suse.initrd  
ftp> asc  
ftp> get /suse/images/parmfile suse.parmfile
```

3. After you have the files on your local VM disk, you can manually boot (IPL) or create an EXEC with the commands for booting. Creating the EXEC was preferred in order to be able to boot several times.

The commands to boot from the local VM disk that were put into the EXEC are:

```
'close rdr'  
'purge rdr all'  
'spool punch * rdr'  
'PUNCH SUSE IMAGE A (NOH'  
'PUNCH SUSE PARM A (NOH'  
'PUNCH SUSE INITRD A (NOH'  
'change rdr all keep nohold'  
'ipl 00c clear'
```

These commands have the following meanings:

```
'close rdr'
```

Closes all open files in the reader so that they can be purged:

```
'purge rdr all'
```

Empties the VM reader. You should ensure that any important reader files have been moved to another location before issuing this command!

```
'spool punch * rdr'
```

Directs the output of the punch device to the reader:

```
'PUNCH SUSE IMAGE A (NOH'
```

Moves the LINUX boot file to the reader:

```
'PUNCH SUSE PARM A (NOH'
```

Moves the LINUX parameters file to the reader:

```
'PUNCH SUSE INITRD A (NOH'
```

Moves the initial RAMDISK file (initial root file system) to the reader:

```
'change rdr all keep nohold'
```

Makes sure the content of the reader is not changed or deleted after the process is finished:

```
'ipl 00c clear'
```

Sends the reader an Initial Program Load (IPL) command, this boots Linux for S/390 in RAMDISK.

After booting you will see the following:

```
=  
=  
==- Welcome to SuSE Linux S/390 7.2 -==  
=  
=
```

First, select the type of your network device:

- 0) no network
- 1) OSA Token Ring

```
2) OSA Ethernet
3) OSA-Express Gigabit Ethernet (experimental)
4) Channel To Channel
5) Escon
6) IUCV (experimental)
Enter your choice (1-6):
```

The choice for our network was 6 IUCV:

```
Please enter your full host name (e.g. s390.suse.com): linux1
Please enter your IP address: 10.0.0.2
Please enter the IP address of your peer: 10.0.0.1
Please enter the IP address of the DNS server (leave blank for none):
9.38.156.250
Please enter the DNS search domain (e.g. suse.com): sanmateo.ibm.com
Please enter the MTU (Maximum Transfer Unit, leave blank for default:
2000):
1492
Configuration for iucv0 will be:
Full host name : linux1.sanmateo.ibm.com
IP address : 10.0.0.2
Peer IP address : 10.0.0.1
DNS IP address : 9.38.156.250
DNS search domain: sanmateo.ibm.com
MTU size : 1492
Is this correct (Yes/No) ? yes
```

After entering the network definitions, a root password is requested.

For security reasons you have to set a temporary installation system password for the user root.

You will be asked for it only when you Telnet into this installation system. This is to limit the access to it, and it will be cleared as soon as you shut down or reset the installation system:

```
Please enter the temporary installation password: oracle
Temporary installation password set.
restarting syslogd:
```

The boot process does network checking, completes the boot process, and then the Linux prompt will appear:

```
SuSE Instsys linux1:/#
```

At this point you can Telnet to Linux and continue the install from a Telnet session with at least 80X25 lines, which is a requirement of YaST. YaST is the utility we used to install Linux.

The first thing you will see when you Telnet will be the following:

```

>>> >>> >>> >>> >>> >>> SuSE Linux S/390 7.2 <<< <<< <<< <<< <<< <<<
1. If you want to check which devices the dasd driver can see, run
'insmod dasd probeonly'
This may take a long time; check the output of 'cat /proc/dasd/devices'.
Remove the dasd driver with 'rmmod dasd' afterwards.
2. Choose the device numbers you want to use for SuSE Linux S/390
!!! BE CAREFUL WHEN SELECTING DASDs - !!!
!!! YOU MAY DESTROY DATA ON SHARED DEVICES !!!
3. Enter
'insmod dasd dasd=<list of devices>'
Remember to separate devices by commas (<dev_no>,<dev_no>),
syntax for ranges is <from_dev_no>-<to_dev_no>
like
'insmod dasd dasd=FD00-FD0F,FD40,FD42,FD80-FD86'
Note: When updating, you have to load the dasd driver with the same
DASD order as in the installed system - see documentation for
further information.
4. Start installation or update with 'YaST'.

```

In order to do the install, we need to tell Linux which DISK devices are allocated to us. Device 306 is the dedicated DISK and 345 is the shared device where we will have our Oracle binaries. As devices were the only ones used for LINUX1, we chose to have VM handle all the paging, therefore, we did not have a swap device.

To let Linux know about these devices, we did the following:

```
insmod dasd dasd=306,345
```

At this point we were ready to run YaST and continue with the Linux install. We followed the method documented in *Linux for zSeries and S/390: Distributions*, SG24-6264, in Chapter 5., "Installing SuSE Linux."

## Cloning Linux

Once the installation of our first Linux was completed, and we tested our installation, we started the process of setting up new Linux machines. We had the choice of installing the same way we did the first install, or we could clone our first Linux machine. We did both.

## Cloning

Copying the root file system from the dedicated disk for LINUX1 to the dedicated disk for LINUX2, ipl LINUX2, and making the network changes using YaST or updating /etc/rc.config, and changes to /etc/zipl.conf accomplished the cloning:

```

Ready; T=0.01/0.01 05:25:25
DDR
VM/ENTERPRISE SYSTEMS ARCHITECTURE DASD DUMP/RESTORE PROGRAM

```

```

ENTER:
SYSPRINT CONS
ENTER:
INPUT 306 DASD
ENTER:
OUTPUT 312 DASD
ENTER:
COPY ALL
HCPDDR711D VOLID READ IS
DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:
YES
HCPDDR716D NO VOL1 LABEL FOUND
DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:
YES
COPYING
COPYING DATA 01/15/02 AT 10.28.05 GMT FROM
INPUT CYLINDER EXTENTS OUTPUT CYLINDER EXTENTS
START STOP START STOP
00000000 00002799 00000000 00002799
END OF COPY

```

When using DDR to copy the filesystem, we had to make sure that our Linux machine was shut down properly, and that the target disk was formatted with the RESERVE parameter:

/etc/rc.config:

```

FQHOSTNAME="linux1.sanmateo.ibm.com"
FROM_HEADER="linux1.sanmateo.ibm.com"
IPADDR_0="10.0.0.2"
IFCONFIG_0="10.0.0.2 pointopoint 10.0.0.1 mtu 1492 up"

```

Change to:

```

FQHOSTNAME="linux2.sanmateo.ibm.com"
FROM_HEADER="linux2.sanmateo.ibm.com"
IPADDR_0="10.0.0.3"
IFCONFIG_0="10.0.0.3 pointopoint 10.0.0.1 mtu 1492 up"

```

/etc/zipl.conf (for linux1):

```

[defaultboot]
default=ipl
[ipl]
target=/boot/zipl
image=/boot/kernel/image
#ramdisk=/boot/initrd
parameters="dasd=0306,0345 root=/dev/dasdb1 noinitrd"
[dumptape]
target=/boot

```

```
dumppto=/dev/rtibm0
```

/etc/zipl.conf (for linux2):

```
[defaultboot]
default=ipl
[ipl]
target=/boot/zipl
image=/boot/kernel/image
#ramdisk=/boot/initrd
parameters="dasd=0312,0345 root=/dev/dasdb1 noinitrd"
[dumptime]
target=/boot
dumppto=/dev/rtibm0
```

## 4.2.5 Installing Oracle9i

At this time there is no formal install process. The Oracle Developer's release was delivered as a tar file, so we followed the release notes of the Oracle 9i developer's release, and installed on the share disk.

This section will be updated when Oracle releases Oracle 9i for Linux on S/390.

## 4.2.6 Oracle experiences

To test the sharing of Oracle binaries, we created a shell script that used the hostname as a unique identifier for the database files and for the ORASID (Oracle system identifier). The shell script stored the files in directory /ora\_db. Each Linux virtual machine had the directory created before the **create** statement was executed. The initi.ora for each instance of Oracle was created before the database **create** script was executed. After we had the prerequisites in place, we executed the shell script and monitored the system for any errors.

The creation of the databases was a success, and we continued to run some basic SQL statements against each instance.

## 4.3 Conclusion

We found that it is possible to share Oracle binaries across Linux virtual machines; the preliminary test showed no problems.





## Migrating Oracle9*i* to Linux for zSeries

Once an application has been selected and sized, the database must be moved from its current platform to the targeted Linux for zSeries image. The process of moving the database to Oracle9*i* for Linux on zSeries is consistent with the process of migrating and/or upgrading any Oracle database, and moving it to another platform.

## 5.1 The process for moving the database

Since Oracle9i is new to Linux for zSeries, and unless the source database is already Oracle9i Release 2, this will be a migration (if the source database is Oracle7) or an upgrade if the source database is Oracle8 or Oracle8i. A migration or upgrade in place is not possible since there is no previous production version of Oracle available for Linux for zSeries.

## 5.2 Overview

There are no special tricks available to upgrade and move Oracle, even if it is currently running in Linux on Intel. The tools available for moving Oracle9i are the standard tools that are available with Oracle9i. For migrations and upgrades, a migration utility is provided, as well as migration and upgrade wizards to facilitate the process. To move data between platforms, the tools generally used are Export, Import, SQL\*Loader or SQL.

The use of SQL\*Loader requires that data first be unloaded into delimited flat files. Oracle does not provide a tool to accomplish this. It must either be done by writing a program to extract table data, or one can use third party tools such as BMC's Fast Unload.

As implied above, there are several methods to move the data. The decision must be made whether to migrate and/or upgrade the database in place, and then to move the data or perform the upgrade as part of moving the data during an import or load using SQL\*Loader.

### 5.2.1 Memory restrictions

The size of database is critical and can be key factor in the ability to run an Oracle9i database in Linux for zSeries. In this case, when we say database size, we are referring to the Oracle9i Systems Global Area, SGA. Since the 31-bit version is the only version available, memory size is limited to 2.1 GBytes, and the size of the SGA is limited to 1 GByte.

The default maximum SGA size on Linux/390 is 750 MB. You can increase the maximum SGA size to 1 GB by changing the SGA base address as follows:

1. Log in as the **oracle** user.
2. Enter the following commands from the `$ORACLE_HOME/rdbms/lib` directory:

```
genksms -s 0x45000000 > ksms.s
make -f ins_rdbms.mk ksms.o
```

```
make -f ins_rdbms.mk ioracle
```

### 3. Start Oracle9i.

During the sizing process, the memory requirements should have been discussed. If it was overlooked, and the database being moved is running on a 64-bit operating system, you should insure that the SGA is a size such that it will fit on the 31-bit version of Oracle9i.

**Note:** The `genksms` command is missing from the CD-ROM for 9.2.0.1, but it is included in the 9.2.0.2 patch set.

## 5.2.2 Database size: data files

The size of the physical database (data files, redo logs, control files) may determine several factors about the move of the database:

- ▶ Size determines storage needs. For instance, the maximum storage you may need is four times the amount of storage defined in the source database. The storage needed in this case would comprise the storage in the source database; the storage needed on the source system to hold the exported data, the storage needed on the target system to hold the exported data; and the storage needed for the target database. This can be reduced by three times the source database if you import from the source platform using named pipes to the target database on the target platform.
- ▶ Size may also determine the complexity of the move. The easiest way, and the one most likely used, will be exports and imports. At the opposite end of the spectrum, you might use a *piece parts* method. First, install the Oracle9i database. Then install the schema (stored procedures, views, etc.) from your libraries, and finally move the data via export/import or unload/load. Depending on the size or complexity of the database, this option may be attractive as all the objects are upgraded as they are installed in the database, and this can be done independent of the source database at a time convenient to the user.
- ▶ The larger the database, the more time it may take to move the data. As technology improves, and as Oracle makes improvements in its tools, this time becomes less and less. However, should the database be in the 100s of GBytes or in TBytes, the time to move the data could still be substantial.

### 5.2.3 Compatibility issues

In general, Oracle has always provided downward compatibility of its products. This policy holds true with Oracle9*i*. Applications written with Oracle7, Oracle8, or Oracle8*i* do not need to be rewritten unless it is intended to take advantage of new function available in Oracle9*i*.

Oracle forms written for earlier releases or the Oracle database do not need to be rewritten, and will run as is. However, it may be advisable to look at new functions provided in Oracle9*i* and determine if this may be advantageous.

There are some specific things that should be checked when moving to Oracle9*i* Release 2. These are some common items and generally checked for every upgrade. They are:

- ▶ Check existing code for new reserved words in Oracle9*i*.
- ▶ Clients should be at SQL\*Net V2 or Net8.
- ▶ There are several OCI things to check.
- ▶ Applications that do not use new function do not need to be precompiled and compiled again.
- ▶ Sever Manager went away. All functions of Server Manager are now in SQL\*Plus. Check any scripts that may have used Server Manager as they will need to be changed.

The process of migrating and upgrading, and all the issues that need particular attention are documented in *Oracle9i Migration Guide*, A90191.

### 5.2.4 Database functional issues

Oracle provides downgrade paths as far back as Oracle7 if needed. There are two ways that a current database can be incompatible with previous releases, and therefore, cannot be downgraded. The two ways are:

1. A new feature stores any data on disk that cannot be processed with your previous release (including data dictionary changes).
2. Language incompatibility. A term Oracle defines as an existing feature behaves differently in the new environment as compared to the old environment.

Oracle provides an init.ora parameter COMPATIBLE=*n.n.n* that when used will disable any new function in current database, and operate it as if were a previous level of the database. It will disable all new functions that will cause the database to become incompatible with previous releases. You also will not be able to use any of the new parameter settings.

The *Migration Guide* documents all the new functions that require Oracle9i.

## 5.3 Moving a small database

The example we provide is moving a small database running in Oracle8 (Release 8.0.4) on z/OS to Oracle9i for Linux on zSeries. The database has two tables that were built to perform earlier tests. The `callup_np` table is a non-partitioned table with 8249 rows and the `flt` table has 17 rows. The procedure is as follows:

- ▶ The export of Oracle8 was done on z/OS. We exported using the tables option. Example 5-1 shows the batch JCL and parameters we used for the export.

*Example 5-1 Batch JCL for export*

---

```
000001 //LINEXP   JOB (0000,OR), 'IMPORT V8', CLASS=A,
000002 //                MSGCLASS=X,MSGLEVEL=(1,1),PRTY=15,NOTIFY=&SYSUID
000003 //EXPORT   EXEC PROC=MPM8EXP, PARM='SYSTEM/MANAGER@W:MPM8
000004 //                PARFILE=/DD/PARFILE'
000005 //SYSOUT   DD   SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
000006 //ORA@MPM8 DD   DUMMY
000007 //SYSIN   DD   DUMMY
000008 //SYSERR   DD   SYSOUT=*
000009 //EXPTFL   DD   DSN=DUTCH.LINEXP.DMP,
000010 //                DISP=SHR
000011 //PARFILE  DD   *
000012 DIRECT=N
000013 FILE=/DD/EXPTFL
000014, RANTS=Y
000015 ROWS=Y
000016 CONSTRAINTS=Y
000017 COMPRESS=N
000018 TABLES=(CALLUP_NP,FLT)
000019 /*
```

---

- ▶ You should always have a way to check that the move was successful. One way is to always check the status of all the objects before they are moved. This will alleviate the problem of trying to fix a stored procedure that will not compile, when in fact it was invalid in the source system. To validate this simple move, we issued the SQL in Example 5-2 on the source database. This provided the number of rows in each table. We will check the logs and issue the same SQL after the move.

*Example 5-2 Query to verify rows migrated*

---

```
SQL> select name from v$database;
NAME
-----
MPM8

SQL> select count(*) from callup_np;
COUNT(*)
-----
      8249

SQL> select count(*) from flt;
COUNT(*)
-----
       17
```

---

- ▶ We created an export file DUTCH.LINEXP.DMP. The export file was FTPed to the LINUX1 image. We FTPed the file to the directory /oracle1 on LINUX1. Insure that the export file is transferred in binary mode.
- ▶ Since the export was done at a table level, we needed to create two tablespaces in Linux similar to the tablespaces on the source database. We created tablespaces similar to the ones on the source system. At this level of export, the table definitions are exported, but not the tablespace definitions. We were doing the move as user SYSTEM. The default tablespace for this user is the SYSTEM tablespace. Putting tables there is not good. If a tablespace is not available with the same name as the source database, the default tablespace of the user doing the import is used. We used the DDL in Example 5-3 to create the tablespaces we needed.

*Example 5-3 Create tablespace DDL*

---

```
SQL> create tablespace callup_np
2 datafile '/oracle1/bfrank/oradata/a390/callup_np.dbf' size 30m;
TABLESPACE CREATED

SQL> create tablespace user_data
2 datafile '/oracle1/bfrank/oradata/a390/flt.dbf' size 10m;
TABLESPACE CREATED
```

---

- ▶ In order to perform the import we used the PARFILE option with the imp utility. We created a PARFILE as shown in Example 5-4 with the options we used for the import.

#### Example 5-4 PARFILE for import

---

```
file = /oracle1/linexp.dmp
log = /oracle1/importLog/imp.log
tables = (callup_np, flt)
rows = y
commit = y
```

---

- ▶ The import ran with no errors. We verified this by looking at the log the was created in the /oracle1/bfrank/importLog/imp.log. The contents of the log are in Example 5-5.

#### Example 5-5 Output of import

---

```
Connected to: Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production
```

```
Export file created by EXPORT:V08.00.04 via conventional path
import done in US7ASCII character set and AL16UTF16 NCHAR character set
import server uses WE8ISO8859P1 character set (possible charset conversion)
export client uses WE8EBCDIC1047 character set (possible charset
conversion)
export server uses WE8EBCDIC1047 NCHAR character set (possible ncharset
conversion)
. importing SYSTEM's objects into SYSTEM
. . importing table          "CALLUP_NP"          8249 rows imported
. . importing table          "FLT"              17 rows imported
About to enable constraints...
Import terminated successfully without warnings.
```

---

- ▶ After the import on the target database, we ran the same query to verify that the number of rows exported were imported. The query showed the right number of rows imported, and matched the number of imported rows listed in the imp.log.

## 5.4 Moving a large database

This section describes our experiences moving a 60 GB database from Linux on Intel to Linux on zSeries. The large database is from a Linux on Intel machine was running Oracle8i (Release 8.1.6.2). The amount of disk needed to move this database was 60 GB.

The steps we executed were:

1. Export the database.
2. Compress the file and move it to the zSeries system.
3. Create the init.ora for the new database.
4. Create the database.
5. Execute the import.
6. Set the correct permissions.
7. Compile the invalid objects.
8. Test access to the new database.

### ***Export the database***

The command used to create the export file was:

```
exp sys/change_on_install file=exp.dmp log=exp.log full=Y rows=Y
consistent=Y grants=Y
```

### ***Compress the export file***

After the export we compressed the file to be able to move it quickly over the network with FTP. To compress the file we used **gzip**.

```
gzip exp.dmp
```

### ***Set up the init.ora***

After the export was compressed and moved to the Linux on zSeries machine, we set up the database. The first thing was to create the init.ora file for the large database. This was our init.ora file:

#### ***initLARGEDB.ora***

```
db_name=LARGEDB
db_block_size          = 8192
row_locking = always
nls_date_format  = DD-MON-RR # Is this still true with 11.5 ?
nls_numeric_characters = "."
nls_sort          = binary
nls_language      = american
nls_territory     = america
max_enabled_roles = 140          # Some modules depend on
                                # this feature.

db_files = 1500                                # LARGE
dml_locks          = 500
open_cursors       = 500 # Consumes process memory, unless
                        # using MTS.
enqueue_resources = 5000 # Max. no of concurrent database
locks.
db_file_multiblock_read_count = 32             # LARGE
db_block_buffers = 32000                       #
LARGE
```

```

log_checkpoint_timeout      = 72000 # Checkpoint at least every 20
mins.
shared_pool_reserved_size  = 8388608
shared_pool_size           = 8388608
utl_file_dir = /ora92/tmp
aq_tm_processes = 1
log_checkpoint_interval = 10000000
processes = 2000 #
LARGE
parallel_max_servers = 8 # LARGE
parallel_min_servers = 0
log_buffer = 1048576 #
LARGE
user_dump_dest            = /ora92/udump
background_dump_dest     = /ora92/bdump
core_dump_dest           = /ora92/cdump
max_dump_file_size = 102400 # limit trace file size to 50 Meg each
_trace_files_public      = TRUE
# If using private rollback segments, place lines of the following
# form in each of your instance-specific init.ora files:
# rollback_segments = (name1, name2)
rollback_segments        =
(rbs1,rbs2,rbs3,rbs4,rbs5,rbs6,rbs7,rbs8,rbs9,rbs10)
sort_area_size           = 256000
# Global Naming -- enforce that a dblink has same name as the db it
connects to
global_names = TRUE
# define two control files by default
control_files = (/ora92/control1, /ora92/control2,/ora92/control3)
_system_trig_enabled = TRUE
o7_dictionary_accessibility = TRUE
compatible = 9.0.1
optimizer_features_enable = 9.0.1
event='10932 trace name context level 32768'
event='10933 trace name context level 512'
event='10943 trace name context level 16384'

```

### **Create the new database**

After creating the init.ora we proceeded to create the database with the shell script **createdb.sh**. We had to make sure we had enough disk space in our filesystem. This was our script to create the database:

#### **createdb.sh**

```

#!/bin/ksh
set +vx
export ORACLE_SID=LARGEDB
sqlplus '/as sysdba'<<!
startup nomount pfile=$ORACLE_HOME/dbs/initLARGEDB.ora;

```

```

spool create_db_LARGEDB
create database LARGEDB
logfile '/ora92/LARGEDB_redo01.log' size 200M,
        '/ora92/LARGEDB_redo02.log' size 200M
maxlogfiles 32
maxloghistory 50
datafile '/ora92/LARGEDB_sys01.dbf' size 1200M,
        '/ora92/LARGEDB_sys02.dbf' size 1400M,
        '/ora92/LARGEDB_sys03.dbf' size 1400M maxdatafiles 50
;

create tablespace rbs datafile '/ora92/LARGEDB_rbs01.dbf' size 1400m;
create rollback segment rbs1 tablespace rbs storage (initial 500K next 200k
maxextents unlimited);
create rollback segment rbs2 tablespace rbs storage (initial 500K next 200k
maxextents unlimited);
create rollback segment rbs3 tablespace rbs storage (initial 500K next 200k
maxextents unlimited);
create rollback segment rbs4 tablespace rbs storage (initial 500K next 200k
maxextents unlimited);
create rollback segment rbs5 tablespace rbs storage (initial 500K next 200k
maxextents unlimited);
create rollback segment rbs6 tablespace rbs storage (initial 500K next 200k
maxextents unlimited);
create rollback segment rbs7 tablespace rbs storage (initial 500K next 200k
maxextents unlimited);
create rollback segment rbs8 tablespace rbs storage (initial 500K next 200k
maxextents unlimited);
create rollback segment rbs9 tablespace rbs storage (initial 500K next 200k
maxextents unlimited);
create rollback segment rbs10 tablespace rbs storage (initial 500K next
200k maxextents unlimited);
# shutdown and restart the db
shutdown
exit
!
sleep 60
sqlplus '/as sysdba'<<!
spool create_db_LARGEDB2
startup pfile=$ORACLE_HOME/dbs/initLARGEDB.ora;
create tablespace user_data datafile '/ora92/LARGEDB_data01.dbf' size
1900m;
exit
!

sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB3

```

```

alter tablespace user_data add datafile '/ora92/LARGEDB_data02.dbf' size
1900m;
exit
!
sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB4
alter tablespace user_data add datafile '/ora92/LARGEDB_data03.dbf' size
1900m;
exit
!
sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB5
alter tablespace user_data add datafile '/ora92/LARGEDB_data04.dbf' size
1900m;
exit
!
sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB6
alter tablespace user_data add datafile '/ora92/LARGEDB_data05.dbf' size
1900m;
exit
!
sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB7
alter tablespace user_data add datafile '/ora92/LARGEDB_data06.dbf' size
1900m;
exit
!
sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB8
alter tablespace user_data add datafile '/ora92/LARGEDB_data07.dbf' size
1900m;
exit
!
sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB8
alter tablespace user_data add datafile '/ora92/LARGEDB_data08.dbf' size
1900m;
exit
!
sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB9
alter tablespace user_data add datafile '/ora92/LARGEDB_data09.dbf' size
1900m;
exit
!
sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB10
create tablespace user_idx datafile '/ora92/LARGEDB_idx01.dbf' size 1900m;

```

```

xit
!
sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB11
alter tablespace user_idx add datafile '/ora92/LARGEDB_idx02.dbf' size
1900m;
exit
!

sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB13
alter tablespace user_idx add datafile '/ora92/LARGEDB_idx03.dbf' size
1900m;
exit
!

sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB14
alter tablespace user_idx add datafile '/ora92/LARGEDB_idx04.dbf' size
1900m;
exit
!

sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB15
alter tablespace user_idx add datafile '/ora92/LARGEDB_idx05.dbf' size
1900m;
alter tablespace user_idx add datafile '/ora92/LARGEDB_idx06.dbf' size
1900m;
exit
!

sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB16
alter tablespace user_idx add datafile '/ora92/LARGEDB_idx07.dbf' size
1900m;
alter tablespace user_idx add datafile '/ora92/LARGEDB_idx08.dbf' size
1900m;
alter tablespace user_idx add datafile '/ora92/LARGEDB_idx09.dbf' size
1900m;
exit
!

sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB17
create tablespace temp datafile '/ora92/LARGEDB_temp01.dbf' size 1900m;
--create tablespace q_data datafile '/ora92/LARGEDB_qdata02.dbf' size 40m;
exit
!
wait
sqlplus '/as sysdba'<<! &
shutdown
exit

```

```

!

sqlplus '/as sysdba'<<!
startup pfile=$ORACLE_HOME/initLARGEDB.ora;
spool create_db_LARGEDB18
@/ora92/dbserv/rdbms/admin/catalog
@/ora92/dbserv/rdbms/admin/catproc
@/ora92/dbserv/ord/admin/ordinst
exit
!

sqlplus '/as sysdba'<<!
connect mdsys/mdsys
spool create_db_LARGEDB19
@/ora92/dbserv/md/admin/mdinst
!

sqlplus internal/manager <<!
spool create_db_LARGEDB20
@/ora92/dbserv/ctx/admin/dr0csys.sql CTXSYS SYSTEM TEMP
exit
!
sqlplus ctxsys/ctxsys <<!
spool create_db_LARGEDB21
@/ora92/dbserv/ctx/admin/dr0inst.sql /ora92/dbserv/bin/libctx9.so
!

sqlplus '/as sysdba'<<!
@/ora92/dbserv/javavm/install/initjvm.sql
!

sqlplus system/manager <<!
spool create_db_LARGEDB22
@/ora92/dbserv/sqlplus/admin/pupbld
!
sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB23
@grants_apps.sql
exit

sqlplus '/as sysdba'<<! &
spool create_db_LARGEDB24
create tablespace statspack datafile '/ora92/stats_data.dbf' size 524272K
reuse;
create tablespace statstmp datafile '/ora92/stats_tmp.dbf' size 524272K
reuse;
exit
!

```

### ***Run the import utility***

Once the database was created, we started the import process using the shell script `import.sh` and parameter file `imp.par`. The import ran successfully.

These are the examples of the files we used:

#### ***imp.par***

```
userid=system/manager
full=y
rows=y
buffer=8388608
recordlength=8388608
commit=y
feedback=20000
file=expdat.p
log=impgrants.log
ignore=y
analyze=n
```

#### ***import.sh***

```
rm expdat.p
mknod expdat.p p
cat exp.dmp.gz | gunzip > expdat.p &
imp parfile=v11.par &
```

### ***Set permissions***

Before we could test for the import success, we needed to run the `grant` commands to authorize the users who needed to access the database.

### ***Compile invalid objects***

The way we checked for invalid objects was by running the following SQL:

```
select count(*) from dba_objects where status='INVALID';
```

Then we ran the following SQL to create a `compile_objs.sql`, which we ran to compile the objects:

```
SQL>compile.sql
```

These are the examples of the scripts we ran:

#### ***Compile.sql***

```
spool compile_objs.sql
SELECT 'alter '
      ||
      decode(object_type,'PACKAGE BODY','PACKAGE',object_type)
      ||
      ','
```

```

||
owner
||
', '
||
object_name
||
' compile '
||
decode(object_type, 'PACKAGE BODY', 'BODY;', ';')
FROM all_objects
WHERE status = 'INVALID'

```

### ***compile\_objs.sql***

```

alter VIEW SYS._ALL_REPCOLUMN compile ;
alter VIEW SYS._ALL_REPCOLUMN_GROUP compile ;
alter VIEW SYS._ALL_REPCONFLICT compile ;
alter VIEW SYS._ALL_REPPARAMETER_COLUMN compile ;
alter VIEW SYS._ALL_REPRESOLUTION compile ;
alter PACKAGE SYSTEM.DBMS_SHARED_POOL compile BODY;
alter PROCEDURE SYSTEM.RECORD compile ;

```

### ***Test the new database***

We used two ways to check if the import and compilation were successful. First, we looked at the import log and did not find any errors. The second way was to test if the import was successful by bringing the application that accesses the database up, and we used some of the functions, as well as using automated test. Both methods gave us the information stating it was a success.





## Using RMAN to handle backup and recovery

This chapter conveys the basic concepts and functionality of Recovery Manager (RMAN), a backup and recovery tool that is shipped with your Oracle database. We show examples of how to use some of the features of RMAN. This chapter should not be viewed as a comprehensive guide to Recovery Manager. For additional, detailed information on the generic capabilities of RMAN, consult the *Oracle 9i Backup and Recovery Guide*.

## 6.1 Introduction

There are several ways to backup your database. You can make cold or offline backups of your database by shutting it down and saving its data files to tape or disk. But with this method your database will not be available during the period of your backup. This is the only way to backup your database if you want to use only **os** commands. The other way is to make hot or online backups. To make online backups you have to change your database to the ARCHIVELOG mode. How to do this is described later in this chapter. With RMAN you can make both online and offline backups. No matter which backup type you choose, the output of the backup is a number of files that you have to organize, and this is one of the main benefits of RMAN compared to user-managed backups. RMAN manages these files, and automatically takes care of which files are needed when you restore any part of your database.

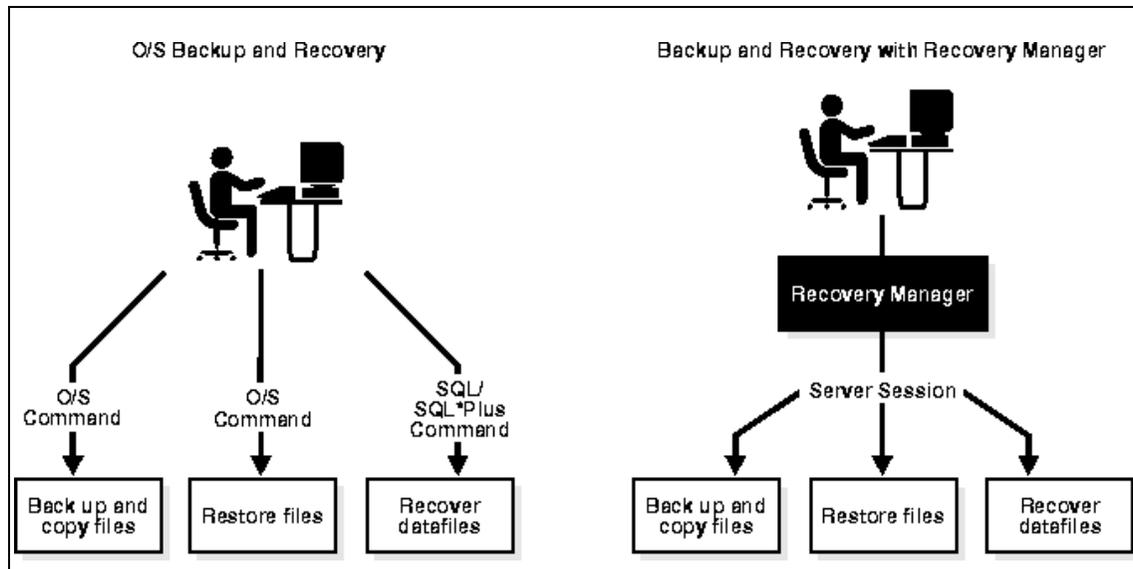


Figure 6-1 Comparison of os backups and RMAN backups

## 6.2 Overview of RMAN

When RMAN begins execution, the following events occur:

RMAN creates two default server sessions that connect to the target database.

If you perform I/O on disk or tape, RMAN requires that you allocate one channel for each disk or device. A channel corresponds to a server session.

If you connect to a Recovery Catalog, RMAN creates a server session on the Recovery Catalog database.

When it connects to your target database, RMAN uses server sessions to perform the backup and recovery operations through a PL/SQL interface. RMAN physically stores its backups and copies on disk, or, if you use media management software, on tape.

Some of the features of RMAN are:

- ▶ Back up and copy the database, TABLESPACES, data files, control files, and archived redo logs
- ▶ Compress backups of data files so that only those data blocks that have been written to are included in a backup
- ▶ Store frequently executed backup and recovery operations in scripts
- ▶ Perform incremental backups, which back up only those data blocks that have changed since a previous backup
- ▶ Create a duplicate of your production database for testing purposes
- ▶ Use third-party media management software
- ▶ Generate a printable message log of all backup and recovery operations
- ▶ Use the Recovery Catalog to automate both restore and recovery operations
- ▶ Perform automatic parallelization of backup and restore operations
- ▶ Restore a backup using a backup control file and automatically adjust the control file to reflect the structure of the restored data files
- ▶ Find data files that require a backup based on user-specified limits on the amount of redo that must be applied for recovery
- ▶ Perform cross checks to determine whether archived materials in the media management catalog are still available
- ▶ Test whether specified backups can be restored

In general there are two ways of running RMAN:

1. The first is to store all information for RMAN in the control file(s). This is the easiest way and the default configuration of RMAN.
2. The second is to store information in a separate database, which is what Oracle recommends.

If you manage several databases in your company, it makes sense to set up a system-wide recovery catalog where you store the backup information for all your databases.

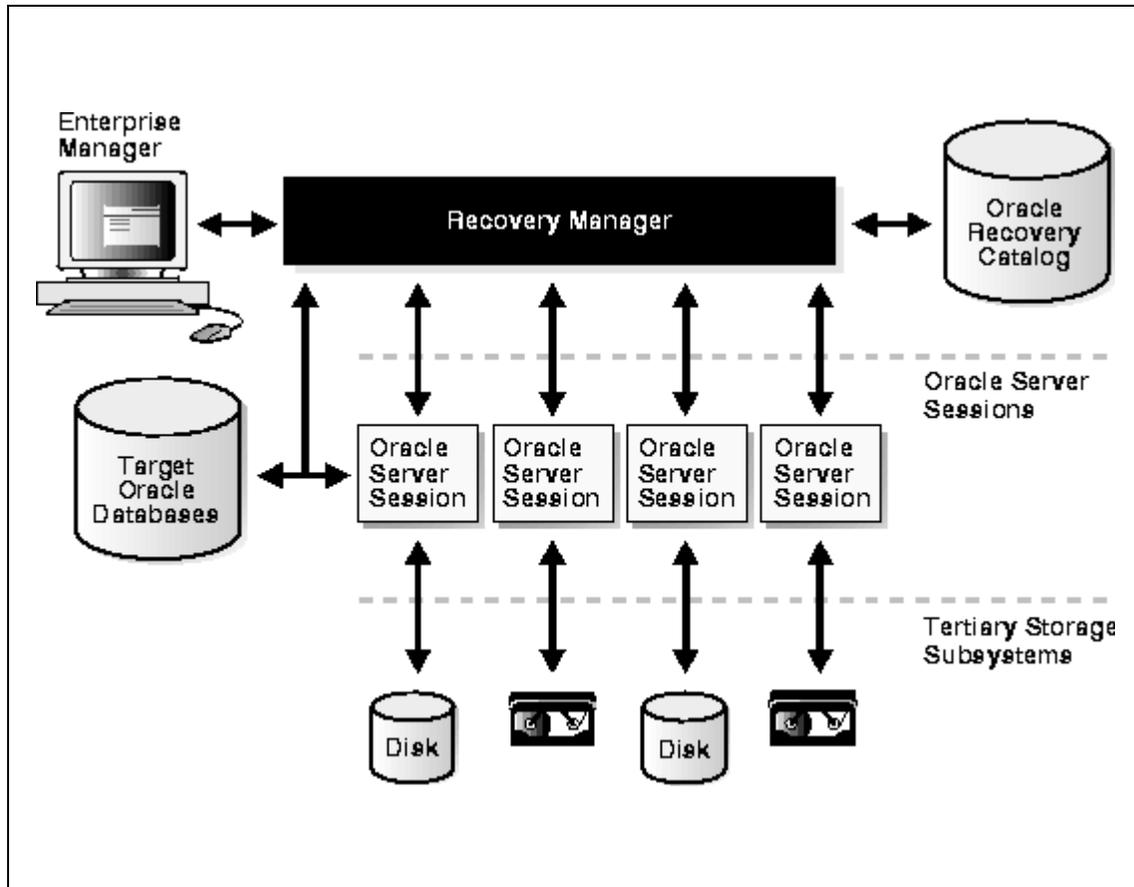


Figure 6-2 Overview of Recovery Manager

## 6.3 RMAN Media Management Interface

For backing up your database to tape, RMAN requires a media manager. RMAN communicates with the media management server through the media management layer (MML). MMLs for other platforms are already available from different vendors with third party tools. You can find a list of vendors on the following Web site:

<http://technet.oracle.com/deploy/availability/htdocs/bsp.htm>

At the time of writing this book, the interfaces for Linux on zSeries from the different vendors were not yet available. So, the only way to back up files to tape is the indirect way, first to disk, and then use tools to back them up to tape. One disadvantage is that you have to manage the restore of the backup files from your tapes. And, you also need temporary disk space to hold your backup files.

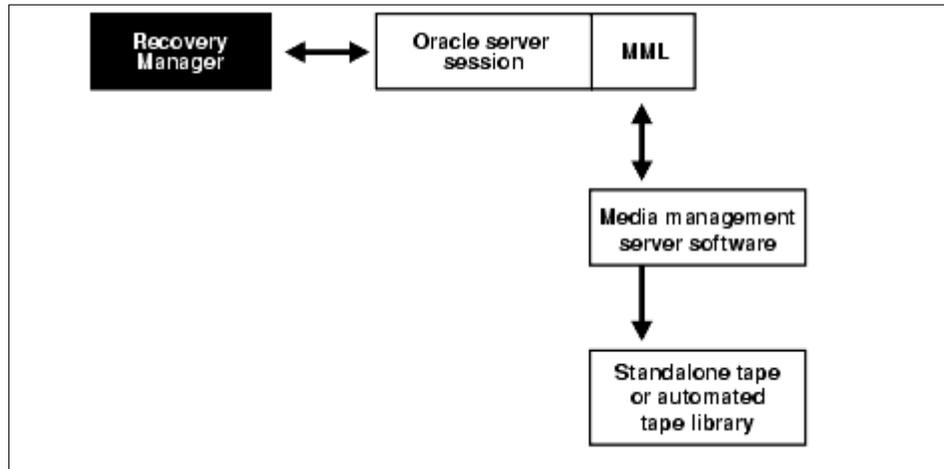


Figure 6-3 Media Management Interface

Picture taken from following URL:

[http://otn.oracle.com/docs/products/oracle9i/doc\\_library/release2/server.920/a96566/rcmintro.htm#430706](http://otn.oracle.com/docs/products/oracle9i/doc_library/release2/server.920/a96566/rcmintro.htm#430706)

## 6.4 Activating archivelog mode

To do an online-backup of the database, it is necessary to run the database in archive log mode. Otherwise, you will get an error message similar to this one:

```

Starting backup at 19-SEP-02
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=12 devtype=DISK
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
including current SPFILE in backupset
including current controlfile in backupset
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 09/19/2002
07:58:34
ORA-19602: cannot backup or copy active file in NOARCHIVELOG mode
  
```

If you decide not to run your database in archivelog mode, and you want to nevertheless back up your database with RMAN, you have to shut down your database first, and then mount it. To do this, you do not have to leave RMAN. There are few commands that cannot only be run within SQL\*Plus, but are also implemented in RMAN.

Examples for these commands are `STARTUP DATABASE`, `SHUTDOWN DATABASE`, or `ALTER DATABASE`. For detailed information of the syntax and what other commands you can use within RMAN, refer to the Oracle documentation of RMAN.

For enabling or disabling the archivelog mode, you only have to mount the database, but do not open it. Afterwards, run the SQL\* Plus command `ALTER DATABASE ARCHIVELOG`:

```
SQL> STARTUP MOUNT
ORACLE instance started.
```

```
Total System Global Area 135340532 bytes
Fixed Size                  451060 bytes
Variable Size               109051904 bytes
Database Buffers            25165824 bytes
Redo Buffers                 671744 bytes
Database mounted.
```

```
SQL> ALTER DATABASE ARCHIVELOG;
```

```
Database altered.
```

Although your database is now in archivelog mode, it still will not archive your log files automatically. To check if automatic archiving is enabled, have a look in the view `v$instance`:

```
SQL> SELECT ARCHIVER FROM v$instance;

ARCHIVE
-----
STOPPED
```

If it is not set (as in this example) you can start it on a running system with the `ALTER SYSTEM` command:

```
SQL> ALTER SYSTEM ARCHIVE LOG START;
```

```
System altered.
```

```
SQL> SELECT ARCHIVER FROM v$instance;
```

```
ARCHIVE
-----
STARTED
```

The changes we made only take effect in memory. After you shut down and restart your database, these changes are lost and automatic archiving is disabled. To enable it after a shutdown, you have to make changes in your startup file. If you start your instance with a pfile, you have to change or insert the line:

```
log_archive_start=true
```

into your pfile. Or, if you start your instance with a spfile you have to use the ALTER SYSTEM SET command within SQL\*Plus. Take care using the SCOPE=SPFILE option. By setting this option, changes are only made in the spfile, but not in memory. Otherwise, you will get an error message, because this parameter cannot be changed while the database is up and running:

```
SQL> ALTER SYSTEM SET log_archive_start=TRUE SCOPE=SPFILE;
```

```
System altered.
```

Verify your changes after the next startup of your database:

```
SQL> SHOW PARAMETERS log_archive_start
```

NAME	TYPE	VALUE
log_archive_start	boolean	TRUE

## 6.5 Using RMAN without a recovery catalog

To start the Recovery Manager, just run the command `rman`, which is stored under the directory `$ORACLE_HOME/bin` after the installation of Oracle. Be sure not to omit the full path of the command, because there is another binary called `rman`, the PolyglotMan. Depending on the way how your `PATH` is set, either one or the other program is run.

The first step in doing a backup with RMAN without a repository is to connect to the database that you want to backup. In our example the instance name is `a390`:

```
oracle@ioc1x:/ > $ORACLE_HOME/bin/rman NOCATALOG
```

```
Recovery Manager: Release 9.2.0.1.0 - Production
```

```
Copyright (c) 1995, 2002, Oracle Corporation. All rights reserved.
```

```
RMAN> CONNECT TARGET sys/ibmoracle@a390
```

```
connected to target database: A390 (DBID=2812401747)
using target database controlfile instead of recovery catalog
```

Now you are connected to the target database, and can perform your backup and restore operations.

**Note:** If you decide to use RMAN without a recovery catalog, keep in mind that it could happen, that RMAN records in the control file are being overwritten. To predefine how long they are kept, set the initialization parameter in the startup file:

```
CONTROL_FILE_RECORD_KEEP_TIME=<days to keep>
```

## 6.6 Using RMAN with a recovery catalog

The steps to implement RMAN with a repository catalog are:

- ▶ Create a user:

```
SQL> CREATE USER rmanager IDENTIFIED BY rman TEMPORARY TABLESPACE temp
      2 DEFAULT TABLESPACE users QUOTA UNLIMITED ON users;
```

```
User created.
```

```
SQL> GRANT RECOVERY_CATALOG_OWNER, CONNECT, RESOURCE TO rmanager;
```

```
Grant succeeded.
```

- ▶ Connect from RMAN to the recovery catalog as the user you created above:

```
oracle@ioczm10s:~ > rman
```

```
Recovery Manager: Release 9.2.0.1.0 - Production
```

```
Copyright (c) 1995, 2002, Oracle Corporation. All rights reserved.
```

```
RMAN> CONNECT CATALOG rmanager/rman@a390
```

```
connected to recovery catalog database
recovery catalog is not installed
```

- ▶ Now create the recovery catalog:

```
RMAN> CREATE CATALOG;
```

```
recovery catalog created
```

- ▶ The last step of setting up RMAN to work with a repository is to register the target database in the repository. To do this you have to connect to both the

repository and the target database. The connect to the repository is done in the prior step, so you only have to connect to the target database:

```
RMAN> CONNECT TARGET sys/ibmoracle@a390

connected to target database: A390 (DBID=2814502571)

RMAN> REGISTER DATABASE;

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete
```

## 6.7 Configuring RMAN

To change the configuration of RMAN, use the command `CONFIGURE` within RMAN. With the command `SHOW ALL` you can view the current configuration:

```
RMAN> SHOW ALL;

RMAN configuration parameters are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; #
default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/ora92/dbserv/dbs/snapcf_a390.f'; #
default
```

## 6.8 Creating and using backup scripts

Within your recovery catalog you can store scripts. To create a backup script in the recovery catalog, you have to be connected to your recovery catalog, and run the `CREATE SCRIPT` command followed by the name of your script:

```
RMAN> CREATE SCRIPT backup_ts_users
2> {
3> ALLOCATE CHANNEL C1 TYPE DISK;
4> BACKUP TABLESPACE users FORMAT='/ora92/backup/users.bkp';
5> }
```

Created script backup\_ts\_users

This script can be invoked by the EXECUTE command of RMAN. A way to start this script from a shell script is shown in Example 6-1. The script should be run as user Oracle.

At the beginning of the script, the profile file of the current user is sourced to set the environment variables such as ORACLE\_HOME. If you have additional or other profile files on your system, add them at the beginning of your script. The lines between the two EOF statements are run as if you would type them on the RMAN prompt. Add there, each backup statement you want to pass to RMAN.

*Example 6-1 Sample shell script to run a RMAN script*

---

```
#####  
#!/bin/bash  
#  
# backupscript backup.sh  
#  
  
. $HOME/.profile  
  
$ORACLE_HOME/bin/rman <<EOF  
CONNECT TARGET sys/ibmoracle@a390 ;  
CONNECT CATALOG rmanager/rman@a390;  
RUN {  
EXECUTE script backup_ts_users;  
}  
EOF  
#####
```

---

You can also run this sample script controlled by the crondaemon to automate your backups. For this script to be run frequently, make an entry in the crontab of the user Oracle.

In Example 6-2 the script is run only on Tuesdays at 3 AM.

*Example 6-2 Sample entry in the crontab*

---

```
0 3 * * 2 /opt/oracle/backup.sh >>/tmp/logfile 2>&1
```

---

## 6.9 Sample BACKUP operations

When you invoke a backup command, RMAN backs up the files into one or more backup sets. You can set parameters for the backup command to specify the filenames for the backup pieces, the number of files to go in each set, and which channel should operate on each input file. Remember, a channel is an Oracle

server process under which an RMAN operation is performed. Files produced with the BACKUP command are in a compressed and special format, which cannot be readily used by an Oracle instance, and must be processed by RMAN before they can be accessed by the server.

*Example 6-3 Sample full database backup*

---

```
RMAN> RUN
1> {
2> ALLOCATE CHANNEL C1 TYPE DISK;
3> BACKUP DATABASE FORMAT='/ora92/backup/fulldb_01.bkp';
4> }
```

```
Starting backup at 23-SEP-02
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
including current SPFILE in backupset
including current controlfile in backupset
input datafile fno=00001 name=/ora92/dbserv/oradata/a390/system01.dbf
input datafile fno=00002 name=/ora92/dbserv/oradata/a390/undotbs01.dbf
input datafile fno=00005 name=/ora92/dbserv/oradata/a390/example01.dbf
input datafile fno=00010 name=/ora92/dbserv/oradata/a390/xdbs01.dbf
input datafile fno=00006 name=/ora92/dbserv/oradata/a390/indx01.dbf
input datafile fno=00009 name=/ora92/dbserv/oradata/a390/users01.dbf
input datafile fno=00003 name=/ora92/dbserv/oradata/a390/cwmlite01.dbf
input datafile fno=00004 name=/ora92/dbserv/oradata/a390/drsys01.dbf
input datafile fno=00007 name=/ora92/dbserv/oradata/a390/odm01.dbf
input datafile fno=00008 name=/ora92/dbserv/oradata/a390/tools01.dbf
channel ORA_DISK_1: starting piece 1 at 23-SEP-02
channel ORA_DISK_1: finished piece 1 at 23-SEP-02
piece handle=/ora92/backup/fulldb_01.bkp comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:57
Finished backup at 23-SEP-02
```

---

In Example 6-3 a complete backup of a database is displayed. Within the run block a channel is allocated for this backup operation. If you do not allocate a channel manually, the default channel is used as configured with the CONFIGURE command. With the FORMAT operator you tell RMAN where to store your backupfile. If you do not specify a filename RMAN stores the backup piece in \$ORACLE\_HOME/dbs/%U. %U is a substitution variable Oracle uses in the format string to generate a unique filename.

*Example 6-4 Sample tablespace backup*

---

```
RMAN> RUN
1> {
2> ALLOCATE CHANNEL C1 TYPE DISK;
3> BACKUP TABLESPACE system FORMAT='/ora92/backup/tb_system_%U.bkp';
```

```

4> }

Starting backup at 25-SEP-02
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
including current SPFILE in backupset
including current controlfile in backupset
input datafile fno=00001 name=/ora92/dbserv/oradata/a390/system01.dbf
channel ORA_DISK_1: starting piece 1 at 25-SEP-02
channel ORA_DISK_1: finished piece 1 at 25-SEP-02
piece handle=/ora92/dbserv/dbs/backup/tb_system_0fe4pi41_1_1.bkp comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:06
Finished backup at 25-SEP-02

```

---

Example 6-4 shows a backup of a single tablespace. Again a channel is allocated manually. In the format string we used the %U substitution variable to generate a unique file name.

## 6.10 Sample RESTORE/RECOVER operations

To restore and recover a complete database, you have to shut it down first, and then mount it. Otherwise, you would get an error message, because RMAN tries to restore the data files to their original location and overwrite existing files. But, these files would be in use by the instance if the database was open.

Example 6-5 shows how to put all required commands into one RUN block.

*Example 6-5 Sample database restore and recover operation*

---

```

RMAN> RUN
1> {
2> SHUTDOWN IMMEDIATE;
3> STARTUP MOUNT;
4> RESTORE DATABASE;
5> RECOVER DATABASE;
6> ALTER DATABASE OPEN;
7> }

database closed
database dismounted
Oracle instance shut down

connected to target database (not started)
Oracle instance started
database mounted

Total System Global Area      135340532 bytes

```

```
Fixed Size                451060 bytes
Variable Size             109051904 bytes
Database Buffers         25165824 bytes
Redo Buffers              671744 bytes
```

```
Starting restore at 23-SEP-02
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=12 devtype=DISK
channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to /ora92/dbserv/oradata/a390/system01.dbf
restoring datafile 00002 to /ora92/dbserv/oradata/a390/undotbs01.dbf
restoring datafile 00003 to /ora92/dbserv/oradata/a390/cwmlite01.dbf
restoring datafile 00004 to /ora92/dbserv/oradata/a390/drsys01.dbf
restoring datafile 00005 to /ora92/dbserv/oradata/a390/example01.dbf
restoring datafile 00006 to /ora92/dbserv/oradata/a390/indx01.dbf
restoring datafile 00007 to /ora92/dbserv/oradata/a390/odm01.dbf
restoring datafile 00008 to /ora92/dbserv/oradata/a390/tools01.dbf
restoring datafile 00009 to /ora92/dbserv/oradata/a390/users01.dbf
restoring datafile 00010 to /ora92/dbserv/oradata/a390/xdm01.dbf
channel ORA_DISK_1: restored backup piece 1
piece handle=/ora92/backup/fulldb_01.bkp tag=TAG20020923T034044 params=NULL
channel ORA_DISK_1: restore complete
Finished restore at 23-SEP-02
```

```
Starting recover at 23-SEP-02
using channel ORA_DISK_1
starting media recovery
media recovery complete
Finished recover at 23-SEP-02
connected to recovery catalog database
```

Database altered

---

If you only want to restore a single tablespace, you do not have to shut down the database. You can set the appropriate tablespace offline, restore, and recover it; and afterwards set the tablespace online. To do this within one RUN block you need to run the sqlplus command `ALTER TABLESPACE` within RMAN. As this command is not implemented in RMAN, you have to use the SQL command. This command runs any SQL statement from within RMAN.

*Example 6-6 Sample tablespace restore and recover operation*

---

```
RMAN> RUN
2> {
3> ALLOCATE CHANNEL C1 TYPE DISK;
4> SQL 'ALTER TABLESPACE users OFFLINE';
5> RESTORE TABLESPACE users;
```

```
6> RECOVER TABLESPACE users;  
7> SQL 'ALTER TABLESPACE users ONLINE';  
8> }
```

```
allocated channel: C1  
channel C1: sid=13 devtype=DISK
```

```
sql statement: ALTER TABLESPACE users OFFLINE  
starting full resync of recovery catalog  
full resync complete
```

```
Starting restore at 26-SEP-02
```

```
channel C1: starting datafile backupset restore  
channel C1: specifying datafile(s) to restore from backup set  
restoring datafile 00009 to /ora92/dbserv/oradata/a390/users01.dbf  
channel C1: restored backup piece 1  
piece handle=/ora92/dbserv/dbs/backup/tb_system_0fe4pi41_1_1.bkp  
tag=TAG20020926T025850 params=NULL  
channel C1: restore complete  
Finished restore at 26-SEP-02
```

```
Starting recover at 26-SEP-02
```

```
starting media recovery  
media recovery complete  
starting full resync of recovery catalog  
full resync complete
```

```
Finished recover at 26-SEP-02
```

```
sql statement: ALTER TABLESPACE users ONLINE  
starting full resync of recovery catalog  
full resync complete  
released channel: C1
```

---

## 6.11 Reporting functionalities of RMAN

The LIST command in RMAN is used to display information from the repository about backup sets and image copies. To run the LIST command, RMAN must be connected to the target database. If you run RMAN in NOCATALOG mode, the database must be mounted. If you use a recovery catalog, RMAN must be connected to it, and the target instance must be started.

The LIST BACKUP SUMMARY command displays all backup pieces belonging to the target database. You will find there, for example, information about the date when the backupset was created, and on which device type it is stored.

*Example 6-7 Summary of all backup sets*

```
RMAN> LIST BACKUP SUMMARY;
```

List of Backups

=====

Key	TY	LV	S	Device Type	Completion Time	#Pieces	#Copies	Tag
39	B	F	A	DISK	23-SEP-02	1	1	TAG20020923T025737
41	B	F	A	DISK	23-SEP-02	1	1	TAG20020923T031830
42	B	F	A	DISK	23-SEP-02	1	1	TAG20020923T034044
43	B	F	A	DISK	25-SEP-02	1	1	TAG20020925T045620
44	B	F	A	DISK	25-SEP-02	1	1	TAG20020925T052551
45	B	F	A	DISK	25-SEP-02	1	1	TAG20020925T071350
105	B	F	A	DISK	26-SEP-02	1	1	TAG20020926T025850

If you need information about backups of single objects of your database, run the LIST BACKUP OF command followed by one of the following parameters:

- ▶ DATAFILE
- ▶ TABLESPACE
- ▶ DATABASE
- ▶ CONTROLFILE
- ▶ SPFILE
- ▶ archivelogRecordSpecifier

*Example 6-8*

```
RMAN> LIST BACKUP OF DATAFILE '/ora92/dbserv/oradata/a390/users01.dbf';
```

List of Backup Sets

=====

BS Key	Type	LV	Size	Device Type	Elapsed Time	Completion Time
39	Full		160K	DISK	00:01:49	23-SEP-02
BP Key: 46 Status: AVAILABLE Tag: TAG20020923T025737						
Piece Name: /ora92/backup/offline.dump						
List of Datafiles in backup set 39						
File LV Type Ckp SCN Ckp Time Name						
-----						
9	Full		1031130	23-SEP-02	/ora92/dbserv/oradata/a390/users01.dbf	

BS Key	Type	LV	Size	Device Type	Elapsed Time	Completion Time
41	Full		160K	DISK	00:01:48	23-SEP-02

BP Key: 47 Status: AVAILABLE Tag: TAG20020923T031830  
Piece Name: /ora92/backup/online2.dump  
List of Datafiles in backup set 41  
File LV Type Ckp SCN Ckp Time Name  
-----  
9 Full 1034190 23-SEP-02 /ora92/dbserv/oradata/a390/users01.dbf

---



# Installing and configuring Oracle Enterprise Manager

The purpose of this chapter is to document our experience installing the Oracle Enterprise Manager (OEM), and the Intelligent Agent (IA) on LINUX/390. We used OEM V9.2.0.1.0. This version uses a three tier approach. We chose to install the Oracle Management Server (OMS) on the same Windows 2000 box with which we ran the OEM Console. The OEM Console can be either standalone or use OMS. This chapter will talk only about using OEM with OMS. The OMS can be installed in one of several Web servers including the Oracle Applications Web Server. The OEM Console can then be accessed from a browser such as Netscape or Internet Explorer.

The focus of the chapter is to demonstrate that an Oracle instance in LINUX/390 can be monitored and managed using OEM in the same way you would manage any other instance of Oracle on any platform other than zSeries. It is based on experiences with Oracle9i.

## 7.1 Documentation

There are several documents available for the installation and configuration processes. These are available on the Oracle documentation CD-ROMs or from Oracle Technology Network (OTN). It would be good, although not necessary, to print the documents. They are available in PDF formats that can be printed, or in HTML format that can be viewed with a browser. The documents are:

- ▶ *Oracle for LINUX Installation Guide, A96167-01*
- ▶ *Oracle Intelligent Agent Users Guide, A96676-01*
- ▶ *Oracle for LINUX Administrator's Guide, A97297-01*
- ▶ *Oracle Enterprise Manager Configuration Guide, A96673-01*
- ▶ *Oracle Enterprise Manager Administrator's Guide, A96674-01*

## 7.2 Introduction to Intelligent Agents and OEM

Intelligent Agents (IA) are autonomous as they function without requiring that the console or Management Server run. The agent that services the Oracle instance (or instances) can run when the database is down. This allows the IA to start and shut down instances. The IAs can also perform administrative tasks and run applications without direct intervention from the DBA. They can also detect errors and issue predetermined corrective actions to resolve problems. All job and event messages are queued, enabling the work to take place when the OMS is not active.

The OEM Version 9.2.0.1 is comprised of three tiers:

- ▶ First tier. This tier is comprised of all the client tools such as the EM Console, DBA Management Pack, Performance Pack, and other integrated tools. The tools can only be launched from a Windows95/98/NT/2000 client or from Sun Solaris. The client can also be run from a Web browser and all the applications can be downloaded from the OMS as Java applets.
- ▶ Middle tier. The second tier is comprised of the Oracle Management Server. The OMS must be installed on either NT/2000 or Sun Solaris. One OMS must be configured in order to use the Enterprise Manager console, and to use the job and event system. OMS provides Access to the Repository. The Repository holds information about the instances being:
  - Managed
  - Event monitoring
  - Request dispatching
  - Event notification (e-mail or pager)
  - Multiple OMSs can be used to share and balance workloads

- ▶ Third tier. This is the Intelligent Agent. It must be installed on all servers that are to be managed by OEM. Managed objects include network nodes (servers), databases, listeners, and other Oracle services.

All three tiers can exist on the same machine, or spread over any number of separate machines in essentially any combination.

We installed the OMS and EM Console (tiers 1 and 2) on a single Windows 2000 workstation, and the IA on the LINUX/390 server. In this configuration we can manage an instance of Oracle9i (release 9.2.0.1) on LINUX/390.

## 7.3 Installing and configuring an OEM configuration

There are three major steps that we did to install OEM, so that it will support an instance on LINUX/390:

1. First we installed an Oracle9 database on Windows 2000.
2. Then we installed and configured the OEM on Windows 2000 and insured everything worked to this point.
3. Afterwards, we installed Oracle9i on the LINUX/390, including the Intelligent Agent.

There are basic requirements that need to be met when installing OEM to support your Oracle databases:

- ▶ A database is needed for the repository.
- ▶ Intelligent Agents must be correctly configured and running on the servers with the databases you want to monitor and administer.
- ▶ OEM must be installed and correctly configured.
- ▶ A dedicated IP address for the machine that will be running OM server is recommended. This is particularly important if you use DHCP. Request that the same IP address be provided through the DNS server for the machine name that is running OEM.

We executed the following steps to install and configure all the components needed in order to support an instance of Oracle on LINUX/390:

1. Install Oracle 9.2 on Windows 2000.  
The agent in this environment is installed with the database.
2. Install OEM in Windows 2000:
  - a. Configure and create a repository.
  - b. Insure that the autodetect works for the local Oracle instance.

3. Install the Intelligent Agent on LINUX/390:
  - a. Install with the database on LINUX/390.
  - b. Insure that the root.sh is run on the LINUX/390 machine as root.
  - c. Check permissions for the user ID the agent runs under.
4. Use OEM to connect and manage the node on LINUX/390.

## 7.4 Installing and configuring an IA on Windows 2000

Our first step was to install an instance of Oracle on Windows 2000. This is on the same machine that we will install OEM. This instance will be used for the repository needed by OEM to store data about the instances being managed. The IA is installed during the database installation and runs as a service under Windows 2000.

After Oracle9 was installed on the Windows 2000 machine, we checked to see that the following services were installed and running:

- ▶ OracleOUIHomeAgent - The Intelligent Agent
- ▶ OracleServiceOR92 - This is the Oracle instance. The SID is the default SID OR92.
- ▶ OracleOUIHomeTNSListener - The TNS listener

**Note:** OR92 is the SID of the database we were using. It is appended to the service name.

If these services are not set to AUTOMATIC, it is advisable to do so, unless you do not want the Oracle database to start when the machine is booted.

There are six files that are important and must be configured correctly for the IA to discover the node (database). The files are:

- ▶ listener.ora - Describes the listener, protocols, and ports.
- ▶ tnsnames.ora - Contains all the service names and network parameters needed to connect to any database on the local machine or on the network.
- ▶ sqlnet.ora - Contains parameters needed for Net8.
- ▶ snmp\_ro.ora - Lists the nodes this agents works with. Do not write in this file. Oracle will make the appropriate entries in this file.
- ▶ snmp\_rw.ora - The same as the above file except the user can make changes here. The most common change that may be made is to enable tracing if something does not work.
- ▶ services.ora - This will contain the descriptors needed to contact to the databases that are supported. Oracle creates this file.

All of the above files are in ORACLE\_HOME\Network\Admin except for services.ora, which is in ORACLE\_HOME\Network\Agent.

These files with the exception of snmp\_ro.ora and services.ora are created during the install process from values that you supply through the installation screens. The Network Configuration Assistant helps set up your connections during and after the install. It will even allow you to test the connection.

These steps will insure all the files are configured correctly to enable discovery of the databases on the local machine.

You should check the files after the install is complete. The agent is started during the install process and the snmp\_ro.ora and services.ora are created.

Example 7-1 through Example 7-5 are examples of our files. The first one is the sqlnet.ora file.

*Example 7-1 Sample of sqlnet.ora file*

---

```
#SQLNET.ORA Network Configuration File: D:\OraHome1\network\admin\sqlnet.ora
# Generated by Oracle configuration tools.
  NAMES.DEFAULT_DOMAIN = us.oracle.com
  SQLNET.AUTHENTICATION_SERVICES= (NTS)
  NAMES DIRECTORY_PATH= (TNSNAMES, ONAMES, HOSTNAME)
```

---

The important entry in this file is the parameter names.default\_domain = world. If world is used, then all service names must use world appended to it. For example, if you add an entry in your tnsnames.ora file for a database, and identify it as or92, the entry in tnsnames would be identified by or92.us.oracle.com. Anything can be used here, although world is a common parameter and is the default.

The next file to check is the listener.ora, which is shown in Example 7-2.

*Example 7-2 Sample of the listener.ora file*

---

```
# ORACLE_HOME\Network\Admin\listener.ora
#LISTENER.ORA Network Configuration File:D:\OraHome1\network\admin\listener.ora
# Generated by Oracle configuration tools.
  LISTENER =
    (DESCRIPTION_LIST =
      (DESCRIPTION =
        (ADDRESS_LIST =
          (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROCO))
        )
        (ADDRESS_LIST =
          (ADDRESS = (PROTOCOL = TCP)(HOST = mgorman-us)(PORT = 1521))
        )
      )
    )
```

```

)
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
(SID_NAME = PLSExtProc)
(ORACLE_HOME = D:\OraHome1)
(PROGRAM = extproc)
)
(SID_DESC =
(GLOBAL_DBNAME = or92.us.oracle.com)
(ORACLE_HOME = D:\OraHome1)
(SID_NAME = or92)
)
)
)

```

---

There may be many entries in the default file provided by the installer.

Next, under the SID\_LIST\_LISTENER, is GLOBAL\_DBNAME. This entry provides a global name to refer to the database that is being listened for by this listener. In our case, the SID is or92, so our GLOBAL\_DBNAME = or92.us.oracle.com. The us.oracle.com is what we decided to use from the entry names.default\_domain = us.oracle.com in the sqlnet.ora file. Next, we made sure the SID\_NAME was correct for our instance, or92.

The next file we configured was the tnsnames.ora. Note that this tnsnames file is in the ORACLE\_HOME\Netork\Admin directory. This tnsnames.ora file is used when a **tnsping** command is executed. Example 7-3 is a copy of our tnsnames file.

*Example 7-3 Sample of the tnsnames.ora file*

---

```

#TNSNAMES.ORA Network Configuration File:D:\OraHome1\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

```

```

EXTPROC_CONNECTION_DATA.US.ORACLE.COM =
(DESCRIPTION =
(AADDRESS_LIST =
(AADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
)
(CONNECT_DATA =
(SID = PLSExtProc)
(PRESENTATION = RO)
)
)

OR92.US.ORACLE.COM =
(DESCRIPTION =
(AADDRESS_LIST =

```

```

        (ADDRESS = (PROTOCOL = TCP)(HOST = mgorman-us)(PORT = 1521))
    )
(CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = or92.us.oracle.com)
)
)

ORA9I.US.ORACLE.COM =
(DESCRIPTION =
    (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = vmlinux11)(PORT = 1521))
    )
    (CONNECT_DATA =
        (SERVICE_NAME = ora9i.us.oracle.com)
    )
)

INST1_HTTP.US.ORACLE.COM =
(DESCRIPTION =
    (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = mgorman-us)(PORT = 1521))
    )
    (CONNECT_DATA =
        (SERVER = SHARED)
        (SERVICE_NAME = MODOSE)
        (PRESENTATION = http://HRService)
    )
)
)

```

---

OEM uses the hostname to discover the node, which is the name of the local machine. From there it reads the services.ora definitions to determine what database services are available for managing.

There are three other files that are important to the discovery process:

ORACLE\_HOME\Network\Admin\snmp\_ro.ora

This file will possibly have one entry in before discovery:

ifile = ORACLE\_HOME\Network\Admin\snmp\_rw.ora

This is an imbed file, indicating that it will use the descriptors in the snmp\_rw.ora file. Do not write anything in this file because it is controlled by Oracle.

Example 7-4 is a copy of the complete snmp\_ro.ora file. It should not be updated.

#### Example 7-4 Sample of snmp\_ro.ora file

---

```
snmp.visibleServices = (listener, or92.us.oracle.com)
snmp.shortname.listener = LISTENER
snmp.longname.listener = listener_mgorman-us.us.oracle.com
snmp.configfile.listener = D:\OraHome1\network\admin\listener.ora
snmp.oraclehome.listener = D:\OraHome1
snmp.servicename.listener_mgorman-us.us.oracle.com =
OracleOUIHomeTNSListener
snmp.SID.or92.us.oracle.com = OR92
snmp.oraclehome.or92.us.oracle.com = d:\orahome1
snmp.address.or92.us.oracle.com =
(ADDRESS=(PROTOCOL=BEQ)(PROGRAM=ORACLE)(ARGV0=ORACLEOR92)(ARGS='(DESCRIP
TION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))'))
snmp.remoteaddress.or92.us.oracle.com = (DESCRIPTION=(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = mgorman-us)(PORT = 1521)) )
(CONNECT_DATA=(SID=OR92)(SERVER=DEDICATED)))
ifile = D:\OraHome1\network\admin\snmp_rw.ora
ifile = D:\OraHome1\network\admin\sqlnet.ora
```

---

#### **ORACLE\_HOME\network\Admin\snmp\_rw.ora**

This file will contain information about the listeners to contact for nodes being managed by OEM. This file will be empty until IA has been successfully initialized. Oracle writes the needed information in the file. If there are problems with the IA, there are entries that can be placed in this file in order to turn on tracing. The procedure is in the *Intelligent Agent Users Guide*, A96676.

After discovery the snmp\_rw.ora, Example 7-5 shows our snmp\_rw.ora file.

#### Example 7-5 Sample of snmp\_rw.ora file

---

```
snmp.contact.listener = ""
snmp.index.listener = 1
snmp.contact.or92.us.oracle.com = ""
snmp.index.or92.us.oracle.com = 2
```

---

#### **ORACLE\_HOME\network\Agent\services.ora**

This file will be empty until IA is successfully initialized. Oracle will list all the services that it can manage with OEM. Do not write in this file. It is recreated every time the IA is stopped and started.

With IA configured and running correctly, we went to the next step of our plan and installed OEM and insured it discovered the Oracle instance on the same machine.

## 7.5 Installing the Oracle Enterprise Manager

Installing OEM is a fairly easy task. It is installed from the database CD set, and there is little interaction needed until it is time to create the repository. In preparation of installing OEM there are a few steps that should be completed. We found out about these during the install. Doing them ahead of time makes the installation and configuration process of OEM easier. Here are the recommendations:

Get an IP address for the Windows 2000 machine if you are using DHCP. OEM needs the same IP address for the machine each time it runs. During startup it checks to insure the IP address matches between the local hostname and hostname in the DNS server. The local machine's hostname must agree with the name from DNS server. The entries in the Hosts file for OEM should be as follows:

```
localhost          127.0.0.1          # local id and tcp loopback
```

Here are the steps we went through to configure OEM and OMS on Windows 2000, you should analyze how you want it configured:

1. At the command prompt type:

```
C:\emca
```

2. This will display the welcome screen. Click **Next**.
3. Configuration Operation window will display. Select **Configure local Oracle Management Server** and click **Next**.
4. This brings up the edit or create dialog. Click **Create**.
5. This then displays: Configure Oracle Management Server. Select **Create a new repository** and click **Next**.
6. Panel displays: Create New Repository Options. Select **Custom** and click **Next**.
7. Panel displays: Select Database Location. Select **In another existing database** and click **Next**.
8. Panel displays: Change SID:

```
Global Database Name: or92.us.oracle.com
SID:                  or92
```

Click **OK**.

9. Panel displays: Select Database for Repository:

```
User name: sys
Password:  *****
Service:   mgorman-us.us.oracle.com:1521:or92
Connect as: SYSDBA
```

Click **Next**.

10. Displays Repository Login Information

User name: OEM\_USER  
Password: \*\*\*\*\*  
Confirm Password: \*\*\*\*\*

Click **Next**.

11. Displays Select Repository User Tablespace

Take Defaults and Click **Next**.

12. Displays Select Repository User Tablespace

Use an existing tablespace: or92

Click **Next**.

13. Displays Create Repository Summary. Click **Finish**.

14. Displays Configuration Assistance Progress. When it indicates success click **Close**.

OEM is now configured and ready for use

## 7.6 Installing the Intelligent Agent and Data Gatherer

This is not a two step process with Oracle 9i on LINUX/390; the Data Gatherer has been integrated into the Intelligent Agent. Furthermore, the Intelligent Agent is installed and configured during the database configuration and installation process. When the database is started, so is the Intelligent Agent as well.

The purpose of this chapter is to document our experience installing and configuring the Oracle Intelligent Agent on LINUX/390.

There are two documents available for the installation and configuration processes. It would be good (although not necessary) to print these documents. They are available in PDF formats, which can be printed, or HTML format:

- ▶ *Oracle Intelligent Agent Users Guide, A96676-01*
- ▶ *Oracle opr LINUX Administrator's Guide, A97297-01*

This is a generic guide for Intelligent Agent information, and can be useful when configuring the Intelligent Agent.

## 7.6.1 Introduction to Intelligent Agents and OEM

Intelligent Agents are autonomous; they function without requiring the Oracle Enterprise Manager console or Management Server to be running. The primary communications path from the Oracle Management Server to the database instance is through the Intelligent Agent. The agent that services an Oracle instance (or instances) can run when the database is down. This allows the Intelligent Agent to start and shut down instances. The Intelligent Agents can also perform administrative tasks and run applications without direct intervention from the DBA. It can detect errors and issue predetermined corrective actions to resolve the problem. All job and event messages are queued, enabling the work to take place when the OMS is not active.

## 7.6.2 Installation of the Intelligent Agent

The process to install the IA is much easier now than it was under previous versions of the database. You define most of the parameters for the files used by the IA using the universal installer configuration process. The only things that remain are:

1. Run the root.sh as root when prompted by the installer.

```
su root
cd $ORACLE_HOME
./root.sh
```

Answer the prompts appropriately for your environment. This creates the oratab file in /etc/verify the oratab file and the files created for IA in ORACLE\_HOME/network/admin and \$ORACLE\_HOME/network/agent.

After the install is completed, the agent should be running.

2. Verify that the account that the agent is running under has permissions to execute against the oratab file.

## 7.6.3 Starting the Intelligent Agent

Make sure you are logged on as the Oracle user. After the install you can check that the IA is running by:

```
agentctl status agent
```

This should return a message that ends with:

Agent in running XXXXXXXX, if the agent is running, or:

Could not contact agent. It may not be running

***If it is shut down:***

- ▶ To start the agent if it is not running:

```
agentctl start agent
```

Starting Oracle Intelligent Agent:

Agent started

- ▶ To stop the agent:

```
agentctl stop agent
```

- ▶ This will return a message:

Agent shutdown

You should verify that the files related to the agent have been configured probably as defined previously in the agent Windows 2000 install. These will be in the \$ORACLE\_HOME/network/admin and \$ORACLE\_HOME/network/agent directories.



## Performance and scalability considerations

This chapter discusses the tools on Linux and VM to analyze performance. It also discusses the major issues of running Linux under z/VM, and the importance of tuning z/VM.

## 8.1 Virtual machine scheduling by z/VM

Linux performance in this environment depends a great deal on VM. Suggested tools are discussed in Appendix B, and this chapter discusses the major performance issues we had to deal with. The first problem we encountered was the scheduling of the Linux guests by z/VM.

When running multiple copies on Linux under VM, we discovered that it was very important to understand how z/VM managed the virtual machines. Some performance issues we observed could only be understood when we researched z/VM scheduling. For example, we noticed that occasionally one or more of the Linux machines would *go to sleep* for protracted periods of time, and then suddenly *wake up* and start processing. We discovered that this was a result of z/VM putting the Linux machine on the eligible list. The following section describes how z/VM schedules the multiple virtual machines, and what you can do to control its behavior.

### z/VM SRM

When z/VM has a number of virtual machines to dispatch, it manages access to the real resources of the system while keeping the virtual machines in one of a number of *queues* or lists. The virtual machines are grouped according to their current execution characteristics and resource requirements. The three queues are called the dormant list, the eligible list, and the dispatch list.

The dormant list consists of the logged-on virtual machines that are not currently being considered for dispatching because one of the following:

- ▶ They have no work to do (are idle).
- ▶ They are waiting for the completion of a long event (such as an I/O operation to a tape drive, or a response to a console read on the virtual operator's console).

On receipt of work to do or on completion of the long event, virtual machines in the dormant list enter the eligible list, where they may wait to enter the dispatch list.

The eligible list consists of the logged-on virtual machines that are not currently being considered for dispatching because to do so would cause the multiprogramming capacity of one or more real system resources (such as paging or storage) to be exceeded. The eligible list forms when there is more demand by virtual machines for real system resources than is available.

Therefore, the virtual machines in the eligible list are ready to run and are waiting to enter the dispatch list.

The dispatch list consists of the logged-on virtual machines that are currently in the set of virtual machines being considered for dispatching. That is, the virtual machines in the dispatch list are competing for use of the real processor(s) and all other scheduled system resources.

For more information on the processor scheduling function of z/VM, see *z/VM Performance*, SC24-5999. If you are planning to deploy many Linux machines under z/VM, we recommend you become familiar with this book.

The CP INDICATE (short form IND) command displays some information on the current status of the system. Example 8-1 on page 133 shows an INDICATE command when the system was not especially busy. Note the part of the response containing the labels Q0, Q1, Q2 Q3, and E1, E2, and E3.

The number after the label is the number of virtual machines (users) in each of z/VMs queues for dispatching. Q0 to Q4 are the counts of users in the dispatch list, and E1 to E3 counts the users in the eligible list.

*Example 8-1 CP INDICATE command*

---

```

ind
AVGPROC-008% 13
XSTORE-000001/SEC MIGRATE-0000/SEC
MDC READS-000004/SEC WRITES-000003/SEC HIT RATIO-076%
STORAGE-043% PAGING-0001/SEC STEAL-000%
Q0-00004(00000)                                DORMANT-00109
Q1-00001(00000)                                E1-00000(00000)
Q2-00000(00000) EXPAN-001 E2-00000(00000)
Q3-00029(00000) EXPAN-002 E3-00000(00000)
PROC 0000-006%                                PROC 0001-011%
PROC 0002-009%                                PROC 0003-007%
PROC 0004-009%                                PROC 0005-010%
PROC 0006-010%                                PROC 0007-009%
PROC 0008-011%                                PROC 0009-008%
PROC 000A-008%                                PROC 000B-009%
PROC 000C-009%
LIMITED-00000
R; T=0.01/0.01 14:25:11

```

---

For more information on the INDICATE command, see *z/VM V4R3.0 CP Command and Utility Reference*, SC24-6008.

In this example, there are 29 users in the Q3 queue. The Q3 users are eligible to be dispatched, but are considered heavy resource users. Q1 is for light users, and Q2 medium weight users. The users in Q3 are our Linux guest machines, which range in size from 1 GB to 2 GB in size. The load was light when the command was executed. You can see that the CPU utilization (AVGPROC) is 8%. If all the machines were busy, the total load on storage is high, and z/VM

may have a lot of paging to do. If the paging rate becomes too high, some of these Linux machines will be placed on the eligible list. This has the effect of stopping them for potentially long periods of time. What do you do if your Linux machines get put on the eligible list? Basically, there are three options:

- ▶ Add resource to the z/VM system. either more memory or more LPAR weight, or both.
- ▶ Specify QUICKDSP for the Linux machines.
- ▶ Use the SET SRM command to adjust the z/VM's scheduler parameters.

The following sections discusses these options:

### ***Resources for z/VM***

In our case we were running over 30 Linux guest machines under z/VM. Several were defined with 2 GB central storage, and others had 1 GB specified. If z/VM had enough storage to fit all these machines in storage at once, then we would not have to worry about some machines being parked on the eligible list. It is unreasonable to expect that the z/VM system have 60 GB of storage, and so most z/VM installations will assume some amount of storage overcommitment. This approach is not likely to succeed in many cases.

### ***Quickdsp***

Note that in Example 8-1 on page 133 there are four users in queue Q0. Q0 is a special queue for users that are defined with the QUICKDSP attribute. This attribute is given to a virtual machine by putting a line in the directory entry for the virtual machine that says: `OPTIONS QUICKDSP`. It can also be added with the command `SET QUICKDSP userid ON`.

Users with the QUICKDSP attribute never go to the eligible list. Specifying this for your Linux guests will guarantee that they will always be run. However, if the sum of all the Linux memory significantly exceeds the amount of memory available to z/VM, then there could be a severe storage shortage, and resulting high page rates. This could cause a serious performance problem. The Linux guests would all run however, and none would be put aside to sleep in the eligible list. Note in Example 8-1 on page 133 that there are four users in Q0, that is, four users with QUICKDSP specified.

If you have a few Linux guests under z/VM, then we recommend you use QUICKDSP to keep the users in storage.

### ***z/VM scheduler and SET SRM command***

The z/VM scheduler breaks the users to be scheduled into three groups. E1 (the Eligible List) and Q1 (the Dispatch List) are for users with short-running transactions. E2 and Q2 are for users with medium-running transactions, and E3 and Q3 is for users with long-running transactions. In Example 8-1 on page 133 you can see that there are 29 users in Q3. In fact, Linux machines always appear in Q3.

In Example 8-2 on page 135 you can see the parameters that the z/VM system resources manager (SRM) uses to control the various virtual machines. The key parameter for controlling paging is STORBUF.

By default this parameter is set to a set of three numbers, 125, 105, and 95. This default means that 95% of the storage can be used by users in Q3. Any percentage greater than 100% represents an overcommitment of storage. The 105 indicates that a 5% overcommitment is acceptable for the sum of Q2 and Q3 users. The 125 number indicates that a 25% overcommitment is acceptable for all users in Q1, Q2, and Q3. If the amount of storage required exceeds these numbers, then a user is pushed onto the eligible list, either E1, E2, or E3, and the storage given to the users on the dispatch list.

Because the Linux machines are always in Q3, by default some will be moved to the eligible list when storage demand exceeds 95% of the total. By using the command SET SRM STORBUF 300 300 300 z/VM will allow a 300% overcommitment of storage before moving a Linux machine to the eligible list.

If you have many Linux machines, we recommend that you use this command rather than QUICKDSP.

#### *Example 8-2 CP Query SRM command*

---

```
q srm
IABIAS : INTENSITY=90%; DURATION=2
LDUBUF : Q1=100% Q2=75% Q3=60%
STORBUF: Q1=125% Q2=105% Q3=95%
DSPBUF : Q1=32767 Q2=32767 Q3=32767
DISPATCHING MINOR TIMESLICE = 5 MS
MAXWSS : LIMIT=9999%
..... : PAGES=999999
XSTORE : 0%
R; T=0.01/0.01 14:24:48
```

---

## 8.1.1 Processor usage

To determine the amount of CPU time used by a Linux virtual machine, we found that the `INDICATE USER` command was the most useful. z/VM has the only accurate viewpoint of what resources each guest is consuming, so any measurements need to start there. The goal of z/VM is to provide each guest with the illusion that it has an entire machine all to itself. A necessary result of that is that no guest can possibly tell what real resources it is using, unless it asks z/VM for the information.

Although Linux has commands such as `top` and `vmstat`, which report on resources used by the Linux machine, we found that they assumed they had 100% of the resources of the machine. Because we were running under z/VM, we found the most correct statistics came from z/VM commands, such as `INDICATE USER`. An example of this command is in Example 8-3 on page 136. The `TTIME` is the amount of CPU time used by that processor since logon. For a description of all the fields reported by this command, see *z/VM V4R3.0 CP Command and Utility Reference*, SC24-6008.

### Example 8-3 CP `INDICATE USER` command

---

```
ind user pazxxq03
USERID=PAZXXQ03 MACH=ESA STOR=1G VIRT=V XSTORE=NONE
IPLSYS=DEV 0200 DEVNUM=00026
PAGES: RES=00236377 WS=00235311 LOCK=00001066 RESVD=00000000
NPREF=00065688 PREF=00000000 READS=00478254 WRITES=00502621
XSTORE=000036 READS=03443K WRITES=03942K MIGRATES=499606
CPU 00: CTIME=93:10 VTIME=701:42 TTIME=717:02 IO=379626
      RDR=000000 PRT=036565 PCH=000000
USERID=PAZXXQ03 MACH=ESA STOR=1G VIRT=V XSTORE=NONE
IPLSYS=DEV NONE DEVNUM=00026
PAGES: RES=00236377 WS=00000000 LOCK=00001066 RESVD=00000000
NPREF=00065688 PREF=00000000 READS=00478254 WRITES=00502621
XSTORE=000036 READS=03443K WRITES=03942K MIGRATES=499606
CPU 01: CTIME=93:10 VTIME=558:07 TTIME=589:52 IO=657813
      RDR=000000 PRT=000000 PCH=000000
R; T=0.01/0.01 14:24:44
```

---

For a longer term view of the system performance data, we used the z/VM feature called RealTime Monitor (RTM). IBM RealTime Monitor is a performance monitor that displays statistics about the execution of z/VM. RealTime Monitor is also a diagnostic tool for monitoring, analysis, and problem solving.

### Example 8-4 Sample RTM display

---

```
z/VM CPU2064 SERIAL 0817C9 13G DATE 08/09/02 START 12:25:24 END 12:25:54

<USERID> %CPU %CP %EM ISEC PAG WSS RES UR PGES SHARE VMSIZE TYP,CHR,STAT
SYSTEM 334 334 .00 .00 .00 0 1579 .0 4G ..... 2G SYS,
```

```

PAZXXQ04 105 11 93 130 149 393K 303K .0 75K 100 1536M VMS,DSC,DISP
PAZXXD01 77 17 60 1210 284 249K 249K .0 212K 100 1536M VMS,DSC,DISP

<--- DEVICE ---> <----- DEVICE RDEV DATA -----> <-- MEASUREMENT FACILITY -->
DEV TYPE VOLSER IOREQST SEC %Q %ER R %LK LNK PA %UT ACC FPT DCT CN %CN
0480 OSA 8373 279 .00 .00 1.9 0 1 .00 0 0 0 0 .00
0481 OSA 7989 266 .00 .00 .29 0 1 .00 0 0 0 0 0 .00
1400 3390 LX1400 785 26 .00 .00 .00 1 6 21 8 0 1 7 18
34C6 3390 VM PG18 595 19 .00 .00 .00 0 6 20 10 0 8 1 3.3

<----- CPU STATISTICS -----> <-- VECTOR ---> <STORAGE><XSTORE>
NC %CPU %US %EM %WT %SY %SP XSI %SC NV %VT %OT RSTR %ST PSEC %XS XSEC TTM
->11 10H 223 387 98 392 123 367K 98 0 0 0 0 71 3569 99 3428 0.003
<-.. 776 121 474 324 181 53 223K 97 .. 0 0 0 75 2786 98 3186 0.023
-----<-- 31 LOG ACTIONS INDICATED -->-----+

```

For more information on RTM, see *z/VM V4R3.0 RealTime Monitor Function Level 410*, SC24-6028. The data that RTM reports can be stored in a monitor file for later reporting. The tool that reports on this data is the Performance Reporting Facility (PRF). For more information on PRF, see *z/VM V4R3.0 Performance Reporting Facility Function Level 410*, SC24-6027.

## 8.1.2 Linux storage management and swapping

The problem of determining the amount of storage required for a Linux machine is complicated by the way Linux uses storage in the virtual machine. Linux will use all storage in the machine, either for disk buffers, or for application use. TAR commands for example, tend to use all the storage in the machine during their processing. From *z/VM*'s perspective, this means that the *working set* of storage used by the Linux guest is the full complete size of the machine. In many cases, you can actually get better performance from a Linux virtual machine if it is defined with a smaller memory size.

Obviously, successful operation of an application running in a small virtual machine depends on the application. Reducing the size of a virtual machine and then measuring the performance is an iterative process, and one that has a distressing trial-and-error feel to it.

If Linux runs out of storage, it will attempt to swap some processes out of the machines memory. The file used for swapping is identified by a **swapon** command. For Linux machines under *z/VM*, we recommend you use a *z/VM* facility called VDISK in memory to supply swap space to Linux. This facility provides a disk file, which offers very high performance access because the data exists only in storage. No main storage resources are used until data is written to the file.

In our configuration we allocated 4 GB of VDISK space for each Linux machine. To do this we put a REXX exec in the virtual machine's PROFILE EXEC file. This gets executed at logon time, and defines two disk files that will be used as swap files after Linux boots. The device addresses of 201 and 202 are used in each Linux machine. They are addresses as /dev/dasdb1 and /dev/dasdc1 in the Linux machine.

The commands in Example 8-6 on page 138 must be executed in the Linux machine after boot. We enabled in.rexecd in /etc/inetd.conf and executed the commands through rexec.

*Example 8-5 GETSWAP EXEC added to the Linux machines PROFILE*

---

```
/* REXX GET Swap Space for LINUX guest */
/* Obtains VDISK in Memory and formats it for swap */
address command
'CP DEFINE VFB-512 201 BLK 4194296'
'ACCESS 201 B'
  push 'SWAP1'
  push '1'
'FORMAT 201 B (BLK 4096'
  push '1'
'RESERVE LINUX SWAP B6'
'RELEASE B'
/* Get another one */
'CP DEFINE VFB-512 202 BLK 4194296'
'ACCESS 202 B'
  push 'SWAP2'
  push '1'
'FORMAT 202 B (BLK 4096'
  push '1'
'RESERVE LINUX SWAP B6'
'RELEASE B'
```

---

*Example 8-6 SWAPON commands executed in Linux*

---

```
mkswap /dev/dasdb1
swapon /dev/dasdb1
mkswap /dev/dasdc1
swapon /dev/dasdc1
```

---

*Example 8-7 CP Q ALLOC PAGE command*

---

```
q alloc page
          EXTENT EXTENT  TOTAL  PAGES  HIGH  %
VOLID  RDEV  START    END  PAGES  IN USE  PAGE  USED
-----  ---  -
-----  ---  -
```

VMPAG1	30A8	1	2225	400500	40635	369079	10%
VMPAG2	30AE	1	2225	400500	40989	388024	10%
VMPAG3	34AE	1	2225	400500	40311	358524	10%
VMPAG4	3298	1	2225	400500	34178	316792	8%
VMPAG5	32AD	1	2225	400500	30911	275917	7%
VMPAG6	32CB	1	2225	400500	34873	301851	8%
VMPAG7	34A1	1	2225	400500	40767	363644	10%
VMPAG8	34E5	1	2225	400500	38729	326372	9%
VMPAG9	32C0	1	2225	400500	33884	276606	8%
VMPAGA	34C0	1	2225	400500	39438	327012	9%
VMPAGB	30E5	1	2225	400500	40950	359474	10%
VMPAGC	30BF	1	2225	400500	40573	346725	10%
VMPG13	34C7	1	2225	400500	39927	352098	9%
VMPG14	34C4	1	2225	400500	40864	385673	10%
.							
.							
.							
VMPG35	32DC	1	2225	400500	33824	290754	8%
VMPG36	32DD	1	2225	400500	34911	319535	8%
VMPG37	32DE	1	2225	400500	33424	300273	8%
VMPG38	32DF	1	2225	400500	33686	285113	8%
				-----	-----	-----	
SUMMARY				15219K	1380K		9%
USABLE				15219K	1380K		9%

R; T=0.01/0.01 14:24:58

---

### 8.1.3 Hints for tuning Linux usage

The following are a number of items that pertain to the performance of the Linux system, or Oracle under Linux.

#### Timer patch

Linux has a 100 HZ timer that wakes up the kernel. The timer count is stored in a variable named *jiffies*. This causes overhead in idle Linux systems. There is a patch available to reduce the overhead for idle Linux systems. The patch is called xxxxxxx, and is shipped on the first patch CD shipped by SuSE for SLES 7.

#### RTM reported Diag 44 counts high

Linux uses a Diagnose code 44 when it enters a spin lock when running under z/VM. This allows z/VM to dispatch another virtual processor instead of wasting time spinning. If the number is high, then consider reducing the number of CPs in the Linux image, as a very high value indicates that there is too high an overcommitment of virtual CPs to real CPs.

## **DASD response time**

We observed high response time on the disk files, and a fairly low I/O rate when we ran without striping the logical volumes. See “Disk capacity and logical volumes” on page 22 for a discussion on defining logical volumes.

## **Oracle parameters**

This version of Oracle (Oracle 9i Release 2 (31-bit) for Linux/390) was built without support for asynchronous I/O. To get some support for overlapping I/O operations to the database, you should specify DBWR\_IO\_SLAVES=4 in the init.ora parameter file.

This Oracle is the same as any other Oracle running under a UNIX operating system. So do not forget to use the standard Oracle tools such as STATSPACK to uncover performance problems in any application.

## **z/VM tuning tips**

For more information on tuning z/VM see the following Web sites:

<http://www.vm.ibm.com/devpages/bitner>

and

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



## Using Statspack to collect VMSTAT information

Statspack is a tool that provides a very comprehensive view of performance information in the Oracle database. CPU utilization can provide important information when diagnosing problems within the Oracle database. This chapter will provide a shell script and shows how we used it to view VMSTAT with the Statspack report. We do not include a diagnostic technique. But with a little creativity, it becomes evident that additional information can be collected with Statspack.

## 9.1 Statspack

Statspack is a powerful tool provided with the Oracle database at no charge. It is similar to `utlbstat` and `utlstat`, except that it has the ability to collect more data and format it in a more readable fashion. The core data contained in Statspack provides the ability to diagnose problems in the Oracle Instance, objects in the database, and SQL. Tuning also requires looking at I/O and network performance as well.

To look at the factors that may be affecting performance outside the Oracle instance, tools such `vmstat`, `iostat` and `netstat` can be used. Generally, these are monitored and collected independent of Oracle.

The book *High-Performance Tuning with Oracle Statspack*, ISBN 0-07-222360-X, was published by The Oracle Press. There are two versions of the book to cover Oracle8i and Oracle9i. The book covers everything from installing Statspack to diagnostic techniques.

In one chapter, we provide a shell script for Linux (korn shell) to capture the `vmstat` information, and put it in an Oracle database in the `perfstat` schema. We found this interesting and altered the script, and changed how the data is reported.

## 9.2 VMSTAT

A short discussion of `vmstat` is relevant here. The utility `vmstat` gathers and displays relevant indicators about system performance. This includes information about the run and wait queues, memory usage, paging, and CPU utilization. This footprint changes with each brand of UNIX.

The utility provides output as shown in Example 9-1 on page 142. This was captured by entering:

```
vmstat 2 2
```

This runs the utility at two second intervals and provides two sets of results.

*Example 9-1 VMSTAT Sample output*

---

```
linux1:~ # vmstat 2 2
  procs           memory    swap          io      system          cpu
 r  b  w  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id
 0  0  0  34824 2684  7008 280172  0  0   3  14  0  16  0  1  11
 0  0  0  34824 2684  7008 280172  0  0   0  14  0  27  0  2  97
linux1:~ #
```

---

There was little system activity when this was taken. But, you can see that you can monitor the number of processes running and waiting, and swap space, memory utilization, and system user utilization.

## 9.3 Gathering VMSTAT as part of Statspack

Generally, `vmstat` (as well as `iostat` and `netstat`) are monitored outside of Statspack. A shell script can be built to collect the `vmstat` information simply by running the utility and redirecting the output to a file such as:

```
vmstat 300 100 > /tmp/msg$$
```

This will run the utility every 5 minutes (300/60) and provide 100 lines of output plus two header lines in the file `/tmp/msg$$` (where `$$` is some number).

Again there is a script in the Oracle Press book that gathers `vmstat` information, puts it in a table that the user `perfstat` (schema owner for Statspack objects) owns. However, not including the errors in the shell script (a `$` missing and parameters split across lines), we thought that with a little creativity it could be improved.

The script was written as a `do forever` loop. It would run `vmstat`, startup `sqlplus` read the data into a table created to save the `vmstat` output, and repeat this until the job was cancelled. You then needed to query the table separately from the Statspack output to generate the information. We thought it better to limit the script to a specific length of time, usually connected to the time you are taking snaps with Statspack, and then only to run `sqlplus` when all the statistics were captured; and then insert that data into the table.

### 9.3.1 Installing Statspack

Installing Statspack is a simple task. We created a tablespace for the table, and then ran the `sqlplus` scripts to install the Statspack packages in the database. But, first we created the `perfstat` tablespace as follows:

```
CREATE TABLESPACE perfstat
datafile 'oracle1/dbserv/oradata/a390/perfstat.dbf'
size 500m;
```

Prior to running the `sqlplus` scripts to install Statspack, insure the two scripts `catdbsyn.sql` and `dmbpool.sql` have been run as the `sys` user.

Next run the `sprecreate.sql` script from `sqlplus` as the `system` user. There will be three questions you need to answer about the installation:

1. The `perfstat` default tablespace is `perfstat`.

2. The perfstat temporary tablespace is temp.
3. The table space where Statspack objects are installed is perfstat.

After the script completes, go into \$ORACLE\_HOME/rdbms/admin and look for \*.lis files that were created. The only \*.lis files that should be there are ones created from installing Statspack. Look in these files for ORA messages or errors. There should be none of either.

The install can be checked out by going to sqlplus and entering:

```
SQL > execute statspack.snap
PL/SQL procedure successfully completed
SQL > execute staaspac.snap
PL/SQL procedure completed
SQL > @spreport
```

A Statspack report should be created.

### 9.3.2 The VMSTAT shell script

Next, we created a table in the perfstat schema to store the data we collected. Some thought should be given here as to what you want to collect. You may not want all the vmstat information, or you may want to vary this. The columns may indicate the exact information collected (as we did) or generic names, such as vmstat\_1, vmstat\_2, etc., and then change the script to put the specific data in columns you want; or even make two or three scripts so the number of changes is limited.

We started **sqlplus** with perfstat user. Below is the DDL we used to create the table:

```
drop table stats$vmstat
create table stats$vmstat (
  start_date      date,
  duration        number,
  runque_waits    number,
  page_in         number,
  page_out        number,
  user_cpu        number,
  system_cpu      number,
  idle_cpu        number,
  wait_cpu        number
)
tablespace perfstat
storage (initial 10m
  next 1m
  pctincrease 0)
;
```

We got the script in the book to work and then altered it as shown in Example 9-2.

### Example 9-2 Shell Script to gather VMSTATS

---

```
#!/bin/ksh
# shell script to capture vmstats and place in table stats$vmstat owned by perfstat
# to execute there are two input parameters. these are inputs to vmstat. the first
# is the sample period in seconds the second parameter is the number of samples to
# capture
#
# script first deletes rows in table from previous runs. then
# runs vmstat. at the end of the vmstat run, the data is read from a msg$$ file
# and the rows are inserted into the stats$vmstat table. the msg file is removed
# at the end of the routine

ORACLE_HOME=/oracle1/dbserv
export ORACLE_HOME

PATH=$ORACLE_HOME/bin:$PATH
export PATH

#determine server name
SERVER_NAME=`uname -a|awk '{print $2}'`
typeset -u SERVER_NAME
export SERVER_NAME

#log into database and clean out the stats$vmstat table for this run

$ORACLE_HOME/bin/sqlplus -s perfstat/perfstat@a390<<EOF
delete from perfstat.stats\${vmstat};
EXIT
EOF

#set sample times
SAMPLE_TIME=$1
SAMPLES=$2

#run vmstat

      vmstat ${SAMPLE_TIME} ${SAMPLES} > /tmp/msg$$

# prep data for insertion into table. Note that the header lines and 1st result are
# deleted. Please refer to the vmstat output for your unix flavor to determine
# what statistics you want to capture. Then change the $ vars to match. You would
# also want to create the table with column names to match what you are collecting

cat /tmp/msg$$ | sed 1,3d | awk '{ printf("%s %s %s %s %s %s\n", $1, $8, $9,
$14, $15, $16) }' | while
read RUNQUE PAGE_IN PAGE_OUT USER_CPU SYSTEM_CPU IDLE_CPU
do
```

```

$ORACLE_HOME/bin/sqlplus -s perfstat/perfstat@a390<<EOF
insert into perfstat.stats\vmstat
      values (
            sysdate,
            $SAMPLE_TIME,
            '$SERVER_NAME',
            $RUNQUE,
            $PAGE_IN,
            $PAGE_OUT,
            $USER_CPU,
            $SYSTEM_CPU,
            $IDLE_CPU,
            0
            );
      EXIT
EOF
done
rm /tmp/msg$$

```

---

### 9.3.3 Procedure to run Statspack and shell script

At this point we performed the following procedure to collect some statistics:

1. Log on to sqlplus as perfstat and execute a snap (execute statspack.snap).
2. Start the shell script **mystat.ksh 300 12**. This takes a sample every 5 minutes for on hour.
3. Take an ending snap and run **@spreport** from the perfstat user.

After thinking about this, we came up with a better way. The **@spreport** provides basic information about the snaps, and asks you which ones you want to use for the report. This script then runs **sprepins.sql**. We copied that script into a directory and altered the script as follows in Example 9-3 on page 146 just before the end of the **sprepins.sql** script.

*Example 9-3 Alterations to the sprepins.sql script*

---

```

*****LINES ABOVE NOT SHOWN*****

or      e.ismodified      != 'FALSE'
      or nvl(e.value,0)      != nvl(b.value,0)
      );

-- Collect the vmstats data

```

```

ttitle lef 'VMSTATS data for this report' skip2;

COLUMN start_date HEADING 'Time'
COLUMN server_name HEADING 'Server |Name'
COLUMN runque_waits HEADING 'Runque |Waits'
COLUMN runque_waits FORMAT 9,999
COLUMN page_in HEADING 'P.I.'
COLUMN page_out HEADING 'P.O.'
COLUMN user_cpu HEADING 'CPU |User'
COLUMN system_cpu HEADING 'CPU |Sys'
COLUMN idle_cpu HEADING 'CPU |Idle'
COLUMN wait_cpu HEADING 'CPU |Wait'

select start_date, runque_waits, page_in, page_out,
       user_cpu, system_cpu, idle_cpu, wait_cpu
from stats$vmstat;

prompt
prompt                               End of Report
prompt

```

---

The COLUMN commands were put in to Help with some formatting. Also, since we knew what system we were on, and what the duration of the vmstat intervals were, we did not query these values.

We did the same process as we did before, ran a snap, started the shell script, ran an ending snap, and then ran a report against the two snaps.

Below in Example 9-4 is the last several lines of the Statspack reports. We were able to run the vmstat utility, collect the results in a table owned by perfstat, and have the results presented as part of the Statspack report.

*Example 9-4 Example of Statspack Report with VMSTAT included*

---

```

*****LINES ABOVE NOT
SHOWN*****

undo_retention          10800
undo_tablespace         UNDOTBS1
user_dump_dest          /oracle1/dbserv/admin/a390/udump
-----

```

VMSTATS data for this report

Time	Runque Waits	P.I.	P.O.	CPU User	CPU Sys	CPU Idle
-----						
	CPU Wait					
-----						
08-OCT-02 0	1	0	0	0	1	99
08-OCT-02 0	0	0	0	0	1	99
08-OCT-02 0	0	0	0	0	1	98
08-OCT-02 0	0	0	0	0	1	99
08-OCT-02 0	1	0	0	0	1	99
08-OCT-02 0	0	0	0	0	2	98
08-OCT-02 0	1	0	0	0	1	98
08-OCT-02 0	0	0	0	1	1	98
08-OCT-02 0	0	0	0	0	3	97
08-OCT-02 0	0	0	0	1	1	98
08-OCT-02 0	0	0	0	0	1	99

END OF REPORT

---

## 9.4 Summary

With a little creativity we are able to collect the vmstats data with the Statspack data. This process can be used with netstat or iostat. Although the iostat utility does not come with SuSE Linux, there are open source versions of this that can be installed on your system. Most any system or user statistic, such as using the **ps -ef** command to check processes can be captured, and can be put in a table and printed out with Statspack.



# Implementing RAC on Oracle9i on Linux on zSeries

This chapter describes how we installed Real Application Clusters (RAC) using Oracle Cluster Manager and Oracle9i Database Version 9.2 for Linux for zSeries under VM. Also we describe how we created a database using DBCA (Database Creation Assistant).

Real Application Clusters is an Oracle feature for sharing a database instance between two or more nodes. The reasons for using RAC are:

- ▶ Increased availability
- ▶ Increased scalability

RAC is easy to implement on zSeries because of the existing mainframe infrastructure features such as:

- ▶ Shared DASD is part of z/VM
- ▶ High Speed Interconnect for z/Linux using one of the following:
  - Channel to channel connection
  - ICUV
  - Hyper sockets
  - Ethernet

In this chapter we cover the steps for the complete installation of RAC, which are:

- ▶ Preparation of VM

- ▶ Set up 2 Linux guests
- ▶ Set up raw device file to share
- ▶ Set up the Linux environment
- ▶ Install, configure and start Oracle Cluster Manager
- ▶ Run the Oracle Installer to install Oracle database binaries with RAC
- ▶ Create a database instance

## 10.1 Preparation

### Hardware

We used a z900 running VM version 4.2.

RAC can be deployed in several ways on zSeries:

- ▶ Two or more instances in one LPAR
- ▶ Two or more LPARs in one system, each with an Oracle instance
- ▶ Two or more systems in the same sysplex, each with an Oracle instance

In our case, we used two instances in one LPAR as shown in Figure 10-1.

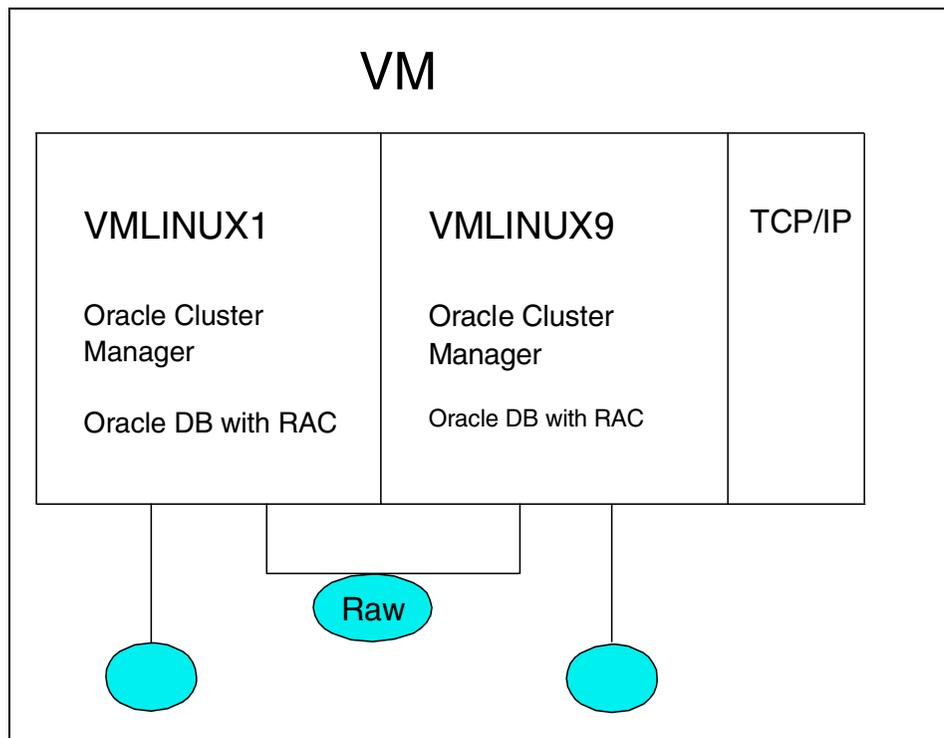


Figure 10-1 Set up for RAC installation

### Software

The environment we used for the installation of Oracle9i was Linux SuSE 7.2 running under VM 4.2.

## Documentation

Oracle documentation can be obtained at:

<http://www.otn.oracle.com/docs/products/oracle9i/content.html>

We used the following Oracle documentation for this installation:

- ▶ *Oracle9i Administrator's Reference Release 2 (9.2.0.1.0) for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris, A97297.* See Appendix F "Oracle Cluster Manager."
- ▶ *Oracle Real Application Clusters Setup and Configuration, A96600*
- ▶ *Oracle9i Installation Guide Release 2 (9.2.0.1.0) for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris, A96167* (see pages 3 - 45 and 3 - 46.)

### 10.1.1 z/VM setup

z/VM provides the capability of running multiple operating systems of the same or of different types on a single mainframe server. These are referred to as *Guest Operating Systems*, which run under VM (virtual machines).

This section shows examples of our:

- ▶ Definitions for two Virtual Linux guests
- ▶ Setup of shared disk for the database
- ▶ Network setup

#### Virtual machine setup

The Linux virtual machines are defined in the user directory of VM. Example 10-1 shows the entries for the machines we used.

*Example 10-1 Definition for Linux guests*

---

```
USER VMLINUX1 xxxxxxxx 512M 1G G
ACCOUNT ITS30000
IPL CMS PARM AUTOCR
MACH ESA 4
POSIXINFO UID 78
DEDICATE 7121 7121
DEDICATE 7122 7122
DEDICATE 7123 7123
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 152 50 VMZU1A MR
MDISK 0201 3390 0001 0200 LX38A0 MR
MDISK 0202 3390 0201 3138 LX38A0 MR
MDISK 0203 3390 3339 3338 LX38A0 MR
MDISK 0204 3390 6677 3338 LX38A0 MR
MDISK 0205 3390 0001 3338 LX3755 MR
MDISK 0206 3390 0001 3338 LX3756 MR
MDISK 0207 3390 0001 3338 LX3757 MR
MDISK 0208 3390 0001 3338 LX25FB MWV
MDISK 0209 3390 0001 3338 LX3A43 MWV
```

```
USER VMLINUX9 xxxxxxxx 512M 1G G
ACCOUNT ITS30000
IPL CMS PARM AUTOOCR
MACH ESA 4
POSIXINFO UID 86
DEDICATE 7124 7124
DEDICATE 7125 7125
DEDICATE 7126 7126
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 1841 50 VMZU1A MR
MDISK 0201 3390 0001 0200 LX3AA4 MR
MDISK 0202 3390 0201 3138 LX3AA4 MR
MDISK 0203 3390 3339 3338 LX3AA4 MR
MDISK 0204 3390 6677 3338 LX3AA4 MR
MDISK 0205 3390 0001 3338 LX2574 MR
MDISK 0206 3390 0001 3338 LX2531 MR
MDISK 0207 3390 0001 3338 LX2537 MR
MDISK 0208 3390 0001 3338 LX25FB MWV
MDISK 0209 3390 0001 3338 LX3A43 MWV
```

---

## Shared disk setup

Oracle RAC requires the sharing of disk between the nodes that will access the database. With IBM mainframes, the sharing of disk is part of the architecture. In our installation we used VM mini disk shared between two virtual machines.

These devices were previously formatted. When devices are not formatted you can use the command **dasdfmt** to do so.

The two lines in Example 10-2 are part of user directory definition for virtual machines vmlinux1 and vmlinux9, which correspond to the share devices we used. These two lines are in each virtual machines definition is shown in Example 10-1 on page 152.

*Example 10-2 Definition for shared disks*

---

```
MDISK 0208 3390 0001 3338 LX25FB MWV
MDISK 0209 3390 0001 3338 LX3A43 MWV
```

---

On the devices defined in Example 10-2 we created logical volumes and raw devices. See “LVM setup” on page 156 and “Raw device support” on page 157 for explanations on how we created logical volumes and raw devices.

We found that we needed to turn off MD CACHE in order to see changes made to the devices from one guest virtual machine when the changes were made on the other guest virtual machine.

In the AUTOLOG1 file (profile exec) on the host VM system (as shown in Example 10-3) we turned off MD CACHE for the shared devices with the following commands.

*Example 10-3 Definition to turn off MDCACHE*

---

```
'CP SET MDCACHE RDEV OFF 25FB'
'CP SET MDCACHE RDEV OFF 3A43'
```

---

## Network setup

The virtual machines we used were configured to use hipersockets for their TCP/IP protocol under VM. Example 10-4 contains the PROFILE TCPIP entries for hipersocket network connectivity.

*Example 10-4 Network definitions*

---

```
DEVICE HIPERDED HIPERS 7100 PORTNAME HIPERPED
LINK HIPERLED QDIOIP HIPERDED
HOME
9.12.9.1 HIPERLED
GATEWAY
9 = HIPERLED 8192 0.255.255.224 0.12.9.0
START HIPERDED
```

---

## 10.1.2 Linux setup

This section assumes you have already installed Linux on all virtual machines where you plan to install Oracle RAC.

We will cover only what is applicable to Oracle RAC installation, which is:

- ▶ Enabling remote shell and remote copy
- ▶ Setting up a raw device

## rsh and rcp functionality

One of the requirements that Oracle RAC installation has is that remote shell and remote copy functionality is in place. In order to have remote shell and remote copy functionality, we updated hosts.equiv files on both machines as shown in Example 10-5.

*Example 10-5 enabling remote shell and remote copy*

---

```
/etc/hosts.equiv for vmlinux1
#
# hosts.equiv   This file describes the names of the hosts which are
#               to be considered "equivalent", i.e. which are to be
#               trusted enough for allowing rsh(1) commands.
# hostname
vmlinux9.itso.ibm.com
vmlinux9
/etc/hosts.equiv file for vmlinux9
#
# hosts.equiv   This file describes the names of the hosts which are
#               to be considered "equivalent", i.e. which are to be
#               trusted enough for allowing rsh(1) commands.
#
# hostname
vmlinux1.itso.ibm.com
vmlinux1
```

---

Linux installation disables remote login by default. To be able to do a remote shell and remote copy, we edited inetd.conf file on both Linux virtual machines as shown in Example 10-6.

*Example 10-6 Example of inet.conf file*

---

```
/etc/inetd.conf modifications for both vmlinux1 and vmlinux9

shell  stream  tcp    nowait  root    /usr/sbin/tcpd  in.rshd -L
login  stream  tcp    nowait  root    /usr/sbin/tcpd  in.rlogind
exec   stream  tcp    nowait  root    /usr/sbin/tcpd  in.rexecd
```

---

**Attention:** Security may be an issue, so make sure you check with your network security specialist to see if you can leave these files in their modified way.

After making changes to the `inetd.conf` file, we needed to restart the network. We used the `rcinetd` command with a restart parameter:

```
vmlinux1:/# rcinetd restart
```

We also edited Oracle user's `.rhosts` file so we could run the `rsh` and `rcp` commands without the need to enter a password. We found that we did not need to have a `.rhosts` file, if the `hosts.equiv` file was updated with the entries for our Linux virtual machines.

*Example 10-7 rhosts file*

---

```
.rhosts
vmlinux9:/ # more ~oracle/.rhosts
vmlinux1
vmlinux9
```

---

We tested the `rsh` and `rcp` commands with the following commands.

*Example 10-8 test rsh and rcp commands*

---

```
oracle@vmlinux1:~ />rsh vmlinux9 df
oracle@vmlinux9:~ rsh vmlinux1 df

oracle@vmlinux1:~ />rcp test vmlinux9:/tmp/test9
oracle@vmlinux9:~ />rcp test vmlinux1:/tmp/test1
```

---

## LVM setup

We used one `dasd` for our raw devices; we divided the `dasd` into many logical volumes using the logical volume manager under `yast`.

The first step was to create a volume group and then logical volumes under that group. The volume group name we used was `rawvg`, and Table 10-1 shows the logical volumes. We created all these logical volumes to be able to accommodate the database we created with DBCA.

*Table 10-1 Logical Volume allocation table*

Logical volume name	Size in K-bytes	Usage
lv_sys	409600	SYSTEM

Logical volume name	Size in K-bytes	Usage
lv_usr	286720	USERS
lv_tmp	40960	TEMP
lv_undo1	225280	UNDO
lv_undo2	225280	UNDO
lv_indx	286720	INDX
lv_tools	32768	TOOLS
lv_ctl1	114688	CONTROL FILE
lv_ctl2	114688	controlfile
lv_redo_1_1	135168	REDO LOG
lv_redo_1_2	135168	REDO LOG
lv_redo_2_1	135168	REDO LOG
lv_redo_2_2	135168	REDO LOG
lv_spfile	8192	SERVER PARAMETER FILE
lv_srvctl	106496	SERVER CONFIGURATION REPOSITORY
lv_quorum	4096	CLUSTER MANAGER NODE MONITOR

## Raw device support

After setting up the logical volumes, we had to install two patches to be able to have raw device support.

The patch numbers are rawio-20020128.rpm and devs-2001.10.11-8.s390.rpm.

We used the `rpm` command with an upgrade parameter on both Linux virtual machines to install the patches.

*Example 10-9 rpm command*

---

```
vmlinux1:/# rpm -U rawio-20020128.rpm
vmlinux1:/# rpm -U devs-2001.10.11-8.s390.rpm
```

---

A requirement to use raw devices is that you bind the raw devices to the logical volumes. We used a shell script to do the binding.

*Example 10-10 rawb command*

---

```
rawbind.sh
raw /dev/raw1 /dev/rawvg1/lv_sys
raw /dev/raw2 /dev/rawvg1/lv_usr
raw /dev/raw3 /dev/rawvg1/lv_tmp
raw /dev/raw4 /dev/rawvg1/lv_undo1
raw /dev/raw5 /dev/rawvg1/lv_undo2
raw /dev/raw6 /dev/rawvg1/lv_indx
raw /dev/raw7 /dev/rawvg1/lv_tools
raw /dev/raw8 /dev/rawvg1/lv_ct11
raw /dev/raw9 /dev/rawvg1/lv_ct12
raw /dev/raw10 /dev/rawvg1/lv_redo_1_1
raw /dev/raw11 /dev/rawvg1/lv_redo_1_2
raw /dev/raw12 /dev/rawvg1/lv_redo_2_1
raw /dev/raw13 /dev/rawvg1/lv_redo_2_2
raw /dev/raw14 /dev/rawvg1/lv_spfile
raw /dev/raw15 /dev/rawvg1/lv_srvctl
raw /dev/raw16 /dev/rawvg1/lv_quorum
```

---

To verify that we had binded the raw devices we ran the **raw -qa** command.

*Example 10-11 command to verify the bind command*

---

```
vmlinux1:/ # raw -qa
/dev/raw1:      bound to major 58, minor 1
/dev/raw2:      bound to major 58, minor 2
/dev/raw3:      bound to major 58, minor 3
/dev/raw4:      bound to major 58, minor 4
/dev/raw5:      bound to major 58, minor 5
/dev/raw6:      bound to major 58, minor 6
/dev/raw7:      bound to major 58, minor 7
/dev/raw8:      bound to major 58, minor 8
/dev/raw9:      bound to major 58, minor 9
/dev/raw10:     bound to major 58, minor 10
/dev/raw11:     bound to major 58, minor 11
/dev/raw12:     bound to major 58, minor 12
/dev/raw13:     bound to major 58, minor 13
/dev/raw14:     bound to major 58, minor 14
/dev/raw15:     bound to major 58, minor 15
/dev/raw16:     bound to major 58, minor 16
```

---

### **Links**

During the creation of the database we found that DBCA did not work with the raw devices, if we did not have a symbolic link to them. We used a shell script to create the symbolic link to the raw devices.

```
rawlink.sh

ln -s /dev/raw1 /oracle1/dbserv/oradata/lv_sys
ln -s /dev/raw2 /oracle1/dbserv/oradata/lv_usr
ln -s /dev/raw3 /oracle1/dbserv/oradata/lv_tmp
ln -s /dev/raw4 /oracle1/dbserv/oradata/lv_undo1
ln -s /dev/raw5 /oracle1/dbserv/oradata/lv_undo2
ln -s /dev/raw6 /oracle1/dbserv/oradata/lv_indx
ln -s /dev/raw7 /oracle1/dbserv/oradata/lv_tools
ln -s /dev/raw8 /oracle1/dbserv/oradata/lv_ct11
ln -s /dev/raw9 /oracle1/dbserv/oradata/lv_ct12
ln -s /dev/raw10 /oracle1/dbserv/oradata/lv_redo_1_1
ln -s /dev/raw11 /oracle1/dbserv/oradata/lv_redo_1_2
ln -s /dev/raw12 /oracle1/dbserv/oradata/lv_redo_2_1
ln -s /dev/raw13 /oracle1/dbserv/oradata/lv_redo_2_2
ln -s /dev/raw14 /oracle1/dbserv/oradata/lv_spfile
ln -s /dev/raw15 /oracle1/dbserv/oradata/lv_srvctl
ln -s /dev/raw16 /oracle1/dbserv/oradata/lv_quorum
```

---

At this point, we had completed the raw devices setup and were ready to continue the install process.

## 10.2 Oracle Cluster Manager install

Oracle RAC is installed as part of Oracle9i Database Version 9.2 creation. This is a change from how Oracle RAC was installed in previous versions.

Before we installed Oracle9i Database Version 9.2, we needed to install the Oracle Cluster Manager for Linux/390. Oracle includes the Oracle Cluster Manager for Linux on zSeries with the distribution of Oracle9i Database version 9.2 for Linux on 390.

Select a path for ORACLE\_HOME that is same on all the chosen nodes and that can be written to from both nodes. Otherwise, installation on the remote nodes will fail. No error message will indicate this failure.

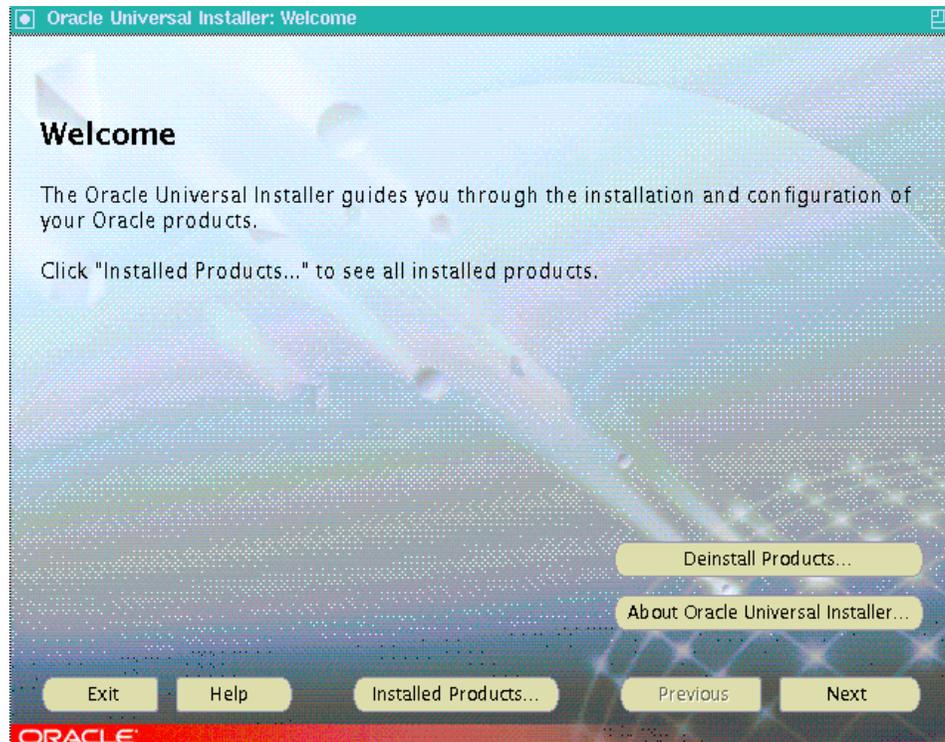
Before we installed Oracle Cluster Manager, we setup several environment variables as shown in Example 10-13. We found it helpful to set up all of them in our Oracle user profile. Oracle's documentation has a complete list of the environment variables that they recommend you set up.

### Example 10-13 env variables

```
export ORACLE_HOME=/oracle1/dbserve
export PATH=$ORACLE_HOME/bin:$PATH
export DISPLAY=9.12.6.154:4
export SRVM_SHARED_CONFIG=/dev/raw15
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib:/usr/local/lib:
/usr/openwin/lib
export DBCA_RAW_CONFIG=$ORACLE_HOME/dbcarawmap
```

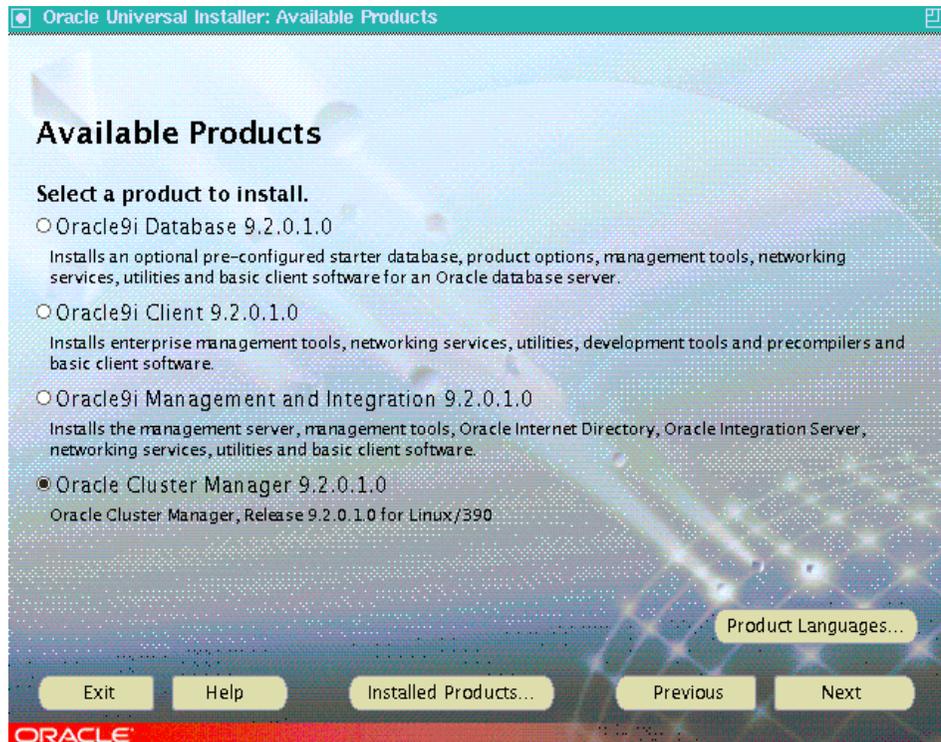
To install Oracle Cluster Manager, we started the Oracle Universal Installer in the same way as if we were installing Oracle9i Database. Refer to Chapter 3., “Installing Oracle9i Database on zSeries with Linux” on page 27.

When you start the Oracle Universal Installer by entering `./runInstaller`, you get the following screen.



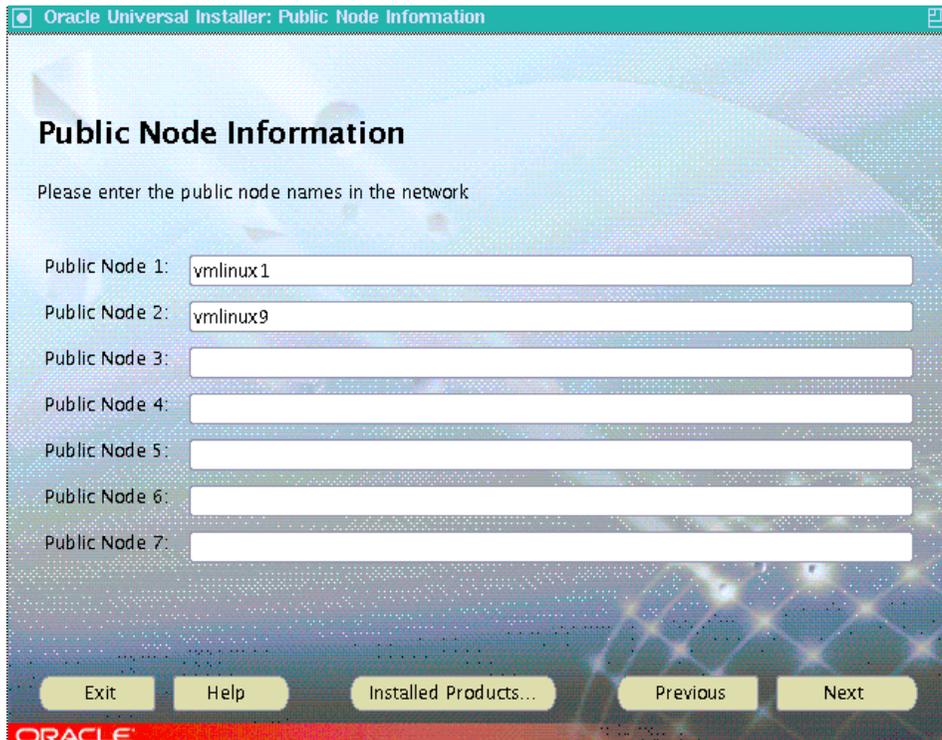
Click the **Next** button to proceed.

Next, you get the product selection screen. Select **Oracle Cluster Manager** and click **Next** to start the installation of the Cluster Manager.



## Public nodes

Oracle Universal Installer prompts for public IP addresses or host names of the nodes in your cluster. These nodes must be available in order for the Oracle Universal Installer to configure the Real Application Clusters properly.

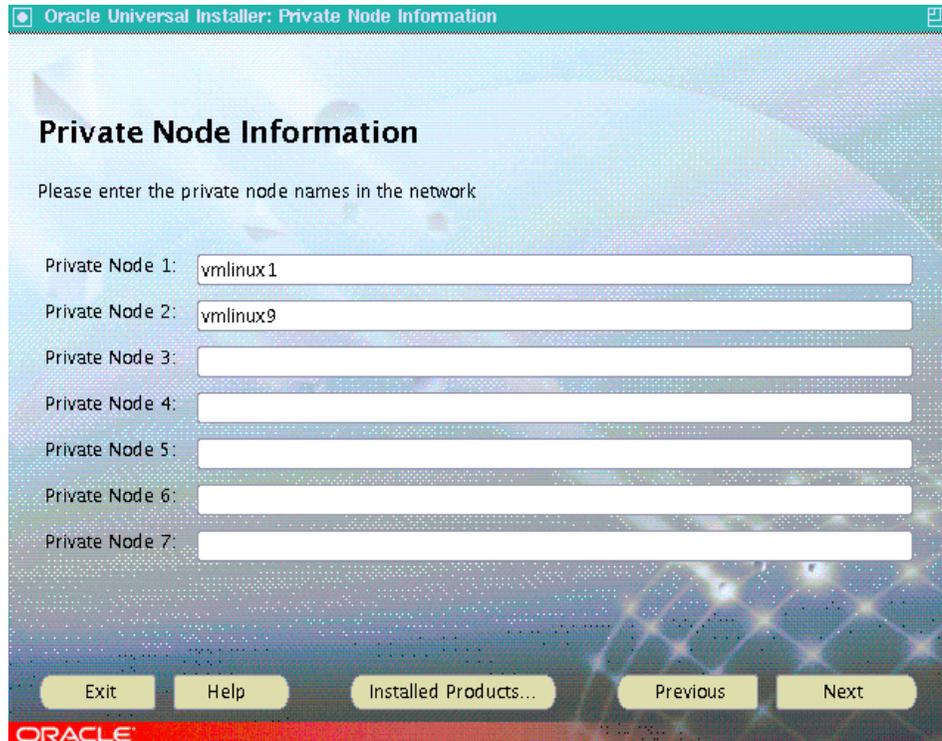


The screenshot shows a window titled "Oracle Universal Installer: Public Node Information". The main heading is "Public Node Information". Below the heading, it says "Please enter the public node names in the network". There are seven input fields labeled "Public Node 1:" through "Public Node 7:". The first field contains "vmlinux1" and the second contains "vmlinux9". The other five fields are empty. At the bottom, there are five buttons: "Exit", "Help", "Installed Products...", "Previous", and "Next". The Oracle logo is visible in the bottom left corner.

In our case, our hosts were VMLINUX1 and VMLINUX9.

## Private nodes

In the same manner as the public nodes, the Oracle Universal Installer prompts for the private nodes. In our installation we used the same nodes for public nodes and private nodes since these were connected by hipersockets and the network connections are handled internally by VM. During the installation and test we did not encounter any problems.

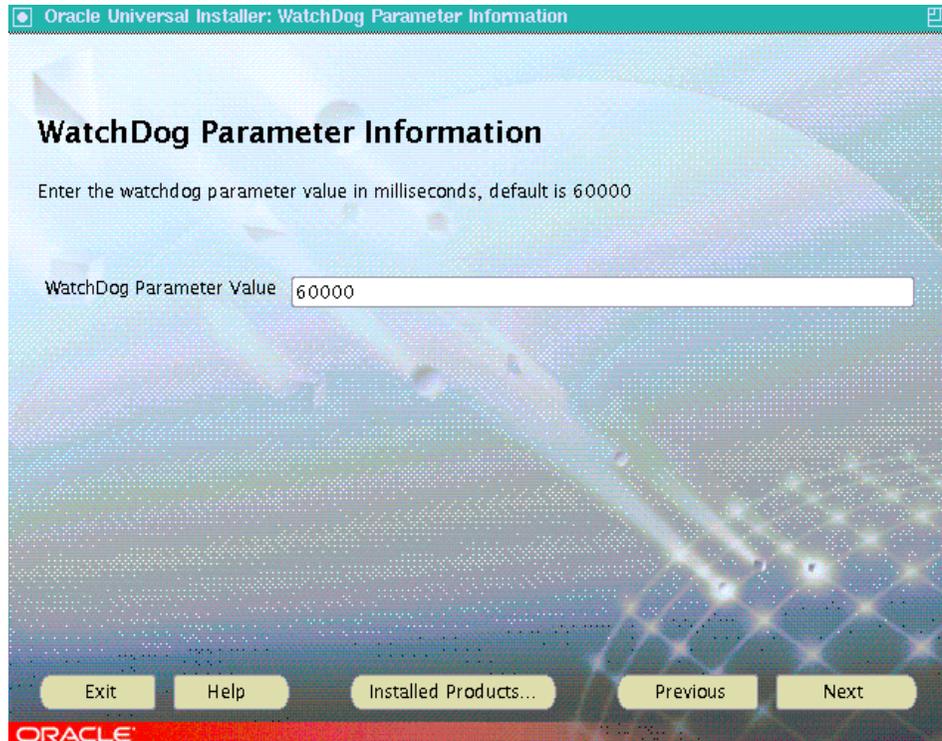


The screenshot shows a window titled "Oracle Universal Installer: Private Node Information". The main heading is "Private Node Information". Below the heading, it says "Please enter the private node names in the network". There are seven input fields labeled "Private Node 1:" through "Private Node 7:". The first field contains "vmlinux1" and the second contains "vmlinux9". The other five fields are empty. At the bottom of the window, there are five buttons: "Exit", "Help", "Installed Products...", "Previous", and "Next". The Oracle logo is visible in the bottom left corner.

Click the **Next** button to proceed.

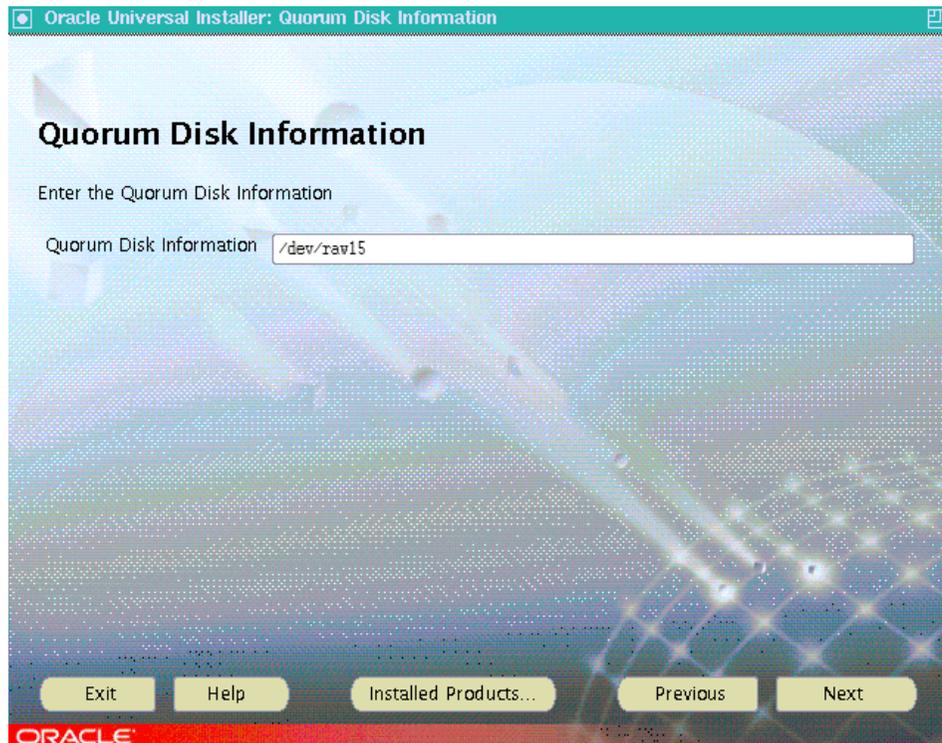
## Watchdog parameter screen

The next screen asks you for the *watchdog* parameters. Simply accept the default value (60000). The zLinux kernel does not support the watchdog device. Oracle Cluster Manager is always started with `/dev/null` as the watchdog device. The Oracle Cluster Manager for Linux on zSeries takes care of the setup and no changes were required.



## Quorum Disk Information

We entered the full name of the raw device we created for the node monitor. This device is referred as the Quorum disk.



## Summary screen

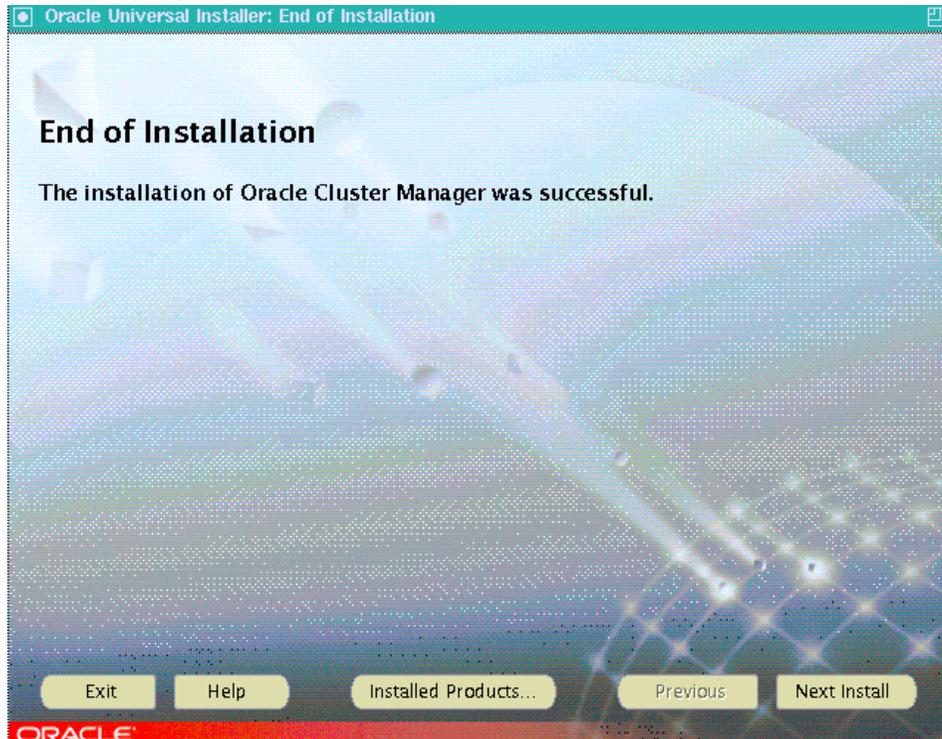
In this screen we saw the components being installed. To continue, we clicked the **Install** button.



## Final screen for Oracle Cluster Manager installation

This screen appeared after the installation of the Oracle Cluster Manager. In one case, this screen came up even if the install did not work properly. We had to logon to the nodes and see if the files were copied to them. We looked in `$ORACLE_HOME` and we saw a directory called `oracm`. This indicated that the `rcp` command worked. To make sure we compared both nodes.

The most common problem was failure of the `rcp` command.



At this point we moved on to continue with the install.

### 10.2.1 Starting Cluster Manager

The next step was to start the Cluster Manager. This service is necessary for the installation of the Oracle9i Database on each node on our cluster. We logged in as root on each node to set up our environment using the Oracle user's profile. See Example 10-13 on page 160. Then we ran the cluster startup script `ocmstart.sh` using the `ocmstart.sh` command:

```
$ORACLE_HOME/oracm/bin/ocmstart.sh
```

This should start the watchdog daemon and the oracm processes which correspond to the Cluster Manager. Verify that it is started on each node, using the `ps` command. Check the messages in the files that are in the log directory at `$ORACLE_HOME/oracm/log`.

When we had problems starting the Cluster Manager, we needed to restart the Cluster Manager. We had to make sure we killed the watchdog process and any oracm processes before we ran the `ocmstart.sh` script again.

## 10.3 Oracle9i install

The Oracle9i install is similar to the install process described in “Installing Oracle9i Database on zSeries with Linux” on page 27. In this chapter we will describe the differences which are minimal.

First we made sure our environment variables were set, as shown in Example 10-13 on page 160. When `ORACLE_HOME` variable was not set we did not get the Cluster Node selection screen when we started the OUI. This screen is where you make the selection of which nodes to install Oracle9i on.

When we did not set the `SVRM_SHARED_CONFIG` variable, the Oracle Universal Installer did not install on the remote nodes, even when `rsh` and `rcp` were working.

We only show the two screens that did not appear in the non RAC installation process.

The only two screens that are different in the RAC install are the selection of the nodes where Oracle9i is to be installed as shown in Figure 10-2 on page 169. This screen will come up after the Welcome screen of the Oracle Universal Installer.

The second screen is the shared configuration filename screen as shown in Figure 10-3 on page 170.

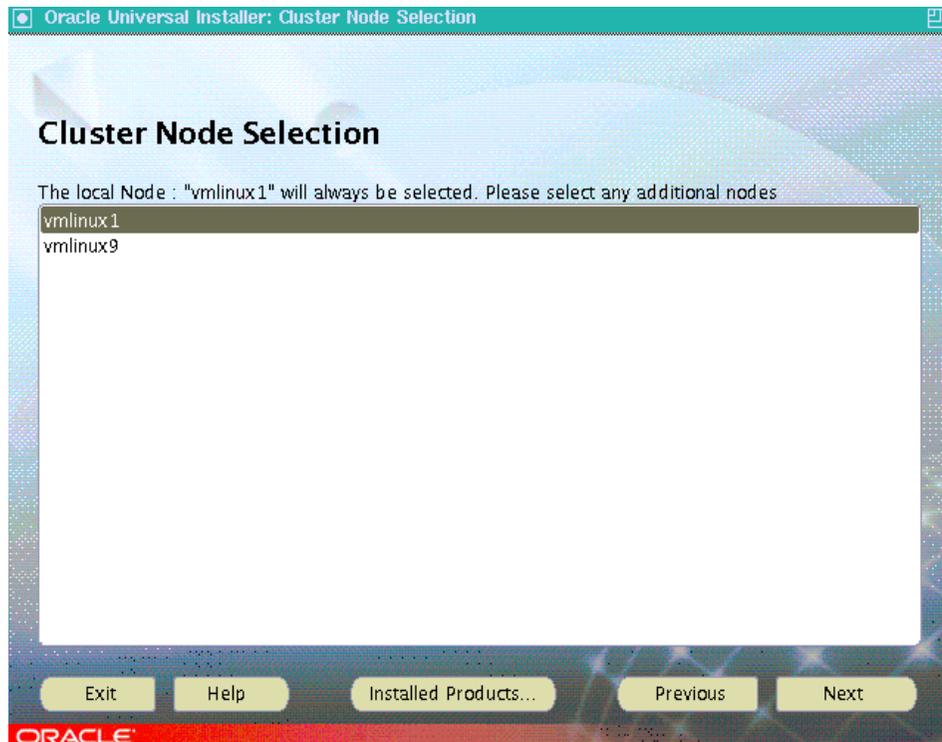


Figure 10-2 Cluster node selection

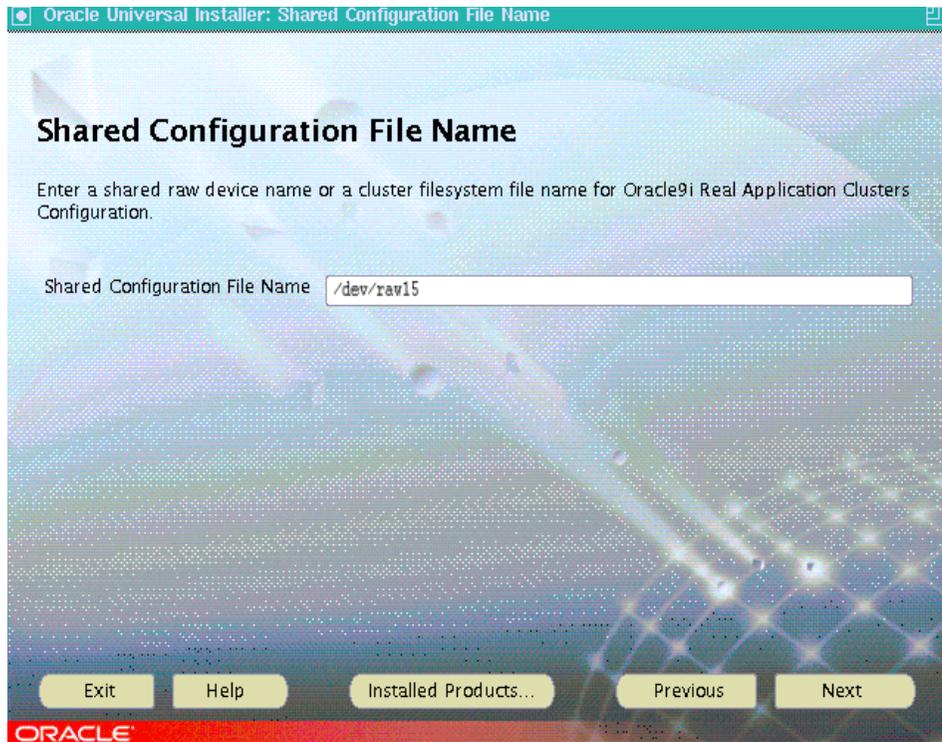


Figure 10-3 Shared Configuration File Name

## 10.4 Database creation

To create a database we used the Oracle's database assistant (DBCA). Before we ran the DBCA utility we had to make sure we had the Global Services Daemon (GSD) running. To check, we ran the `gsdctl stat` (*not status*). There were some times when we had to start the GSD daemon manually. To start the daemon, we ran `gsdctl` with the parameter `start`, as shown in Example 10-14.

### Example 10-14 GSD

```
gsdctl stat
GSD is not running on the local node

gsdctl start
Successfully started GSD on local node

gsdctl ?
```

Usage: gsdctl [options]

where options include:           start       start the gsd  
                          stop        stop the gsd  
                          stat        query the status of the gsd

---

It was very important to allocate the correct amount of space on each logical volume as shown in Table 10-1 on page 156. Each database will need proper planning for space allocation. The values we used were based on the information in Oracle's documentation *Oracle9i Real Application Clusters, Setup and Configuration, Release 2 (9.2), A96600-01*.

Another useful suggestion from the Oracle documentation was to use a map file for DBCA. This was helpful and we found it easier than typing the values in the DBCA interface. To use a DBCA map, you set the environment variable `DBCA_RAW_CONFIG` by issuing the following command:

```
export DBCA_RAW_CONFIG=dbcrawmap
```

Our DBCA map is shown in Example 10-15.

*Example 10-15 DBCA map*

---

```
#Filename dbcrawmap
system=/oracle1/dbserv/oradata/lv_sys
users=/oracle1/dbserv/oradata/lv_usr
temp=/oracle1/dbserv/oradata/lv_tmp
undotbs1=/oracle1/dbserv/oradata/lv_undol
undotbs2=/oracle1/dbserv/oradata/lv_undol2
indx=/oracle1/dbserv/oradata/lv_indx
tools=/oracle1/dbserv/oradata/lv_tools
control1=/oracle1/dbserv/oradata/lv_ctl1
control2=/oracle1/dbserv/oradata/lv_ctl2
redo1_1=/oracle1/dbserv/oradata/lv_redo_1_1
redo1_2=/oracle1/dbserv/oradata/lv_redo_1_2
redo2_1=/oracle1/dbserv/oradata/lv_redo_2_1
redo2_2=/oracle1/dbserv/oradata/lv_redo_2_2
spfile=/oracle1/dbserv/oradata/lv_spfile
```

---

After the creation of the database, we had to make sure that Oracle user on all nodes had permission to write to the raw devices that were allocated for the database.

Our database name was `racz`. The first node `ORACLE_SID` is `rac1`, on the second node the `ORACLE_SID` is `rac2`.

To manage the instances, we used the `svrctl` command as shown in Example 10-16. We had to make sure we had the GSD daemon running on all the nodes. In the example, you see some of the errors when the GSD daemon is not running.

To start only one instance, you use the command `svrctl start instance` and (not the command `svrctl start database`). Oracle's documentation *Oracle9i Real Application Clusters Administration Release 2 (9.2)*, A96596-01, provides the format for the `svrctl` command.

*Example 10-16 svrctl command*

---

```
oracle@vmlinux1:~ > gsdctl ?
Usage: gsdctl [options]
where options include:          start      start the gsd
                                stop       stop the gsd
                                stat       query the status of the gsd
oracle@vmlinux1:~ > gsdctl start
Successfully started GSD on local node

oracle@vmlinux1:~ > svrctl start database -d racz
PRKP-1040 : Failed to get status of the listeners associated with instance
racz2 on nodevmlinux9
PRKC-1019 : Error creating handle to daemon on the node vmlinux9

oracle@vmlinux9:~ > gsdctl start
Successfully started GSD on local node

oracle@vmlinux1:~ > svrctl start database -d racz
oracle@vmlinux1:~ > svrctl status database -d racz
Instance racz1 is running on node vmlinux1
Instance racz2 is running on node vmlinux9

oracle@vmlinux1:~ > svrctl start instance -d racz -i racz2

oracle@vmlinux1:~ > svrctl status database -d racz
Instance racz1 is not running on node vmlinux1
Instance racz2 is running on node vmlinux9
```

---

At this point, we had a database created with RAC enabled. We tested by accessing tables from both instances.



## Using WebSphere with Oracle9i

In this chapter we describe how to install and use IBM WebSphere application server in combination with an Oracle database as the repository database.

## 11.1 Overview

An installation of WebSphere Application Server V4.0 Advanced Edition consists in general of the following components:

▶ **Application server**

The application server runs the application code, such as servlets or EJBs. It works together with the Web server to respond to client requests.

▶ **Administrative server**

The administrative server is running on every node in a WebSphere administrative domain and controls the interaction between the nodes. It is responsible for workload management and transaction coordination.

▶ **Administrative repository**

The administrative repository is a relational database that contains the configuration information for a WebSphere administrative domain.

▶ **Web server and the Web server plug-in for WebSphere**

The Web server responds to client requests by communicating with the application server. The connection between the application server and the Web server is established by Web server plug-ins.

▶ **Administrative console**

The administrative console is a graphical user interface to change the configuration of the application server.

These components do not have to reside on the same machine. Quite the contrary, in most cases it make sense to locate them on different machines. For maintenance reasons for example, it is desirable to store all databases on one platform, so the backup and recovery operations are more cost-effective. For more information about topology selection of WebSphere, refer to *IBM WebSphere V4.0 Advanced Edition Scalability and Availability*, SG24-6192.

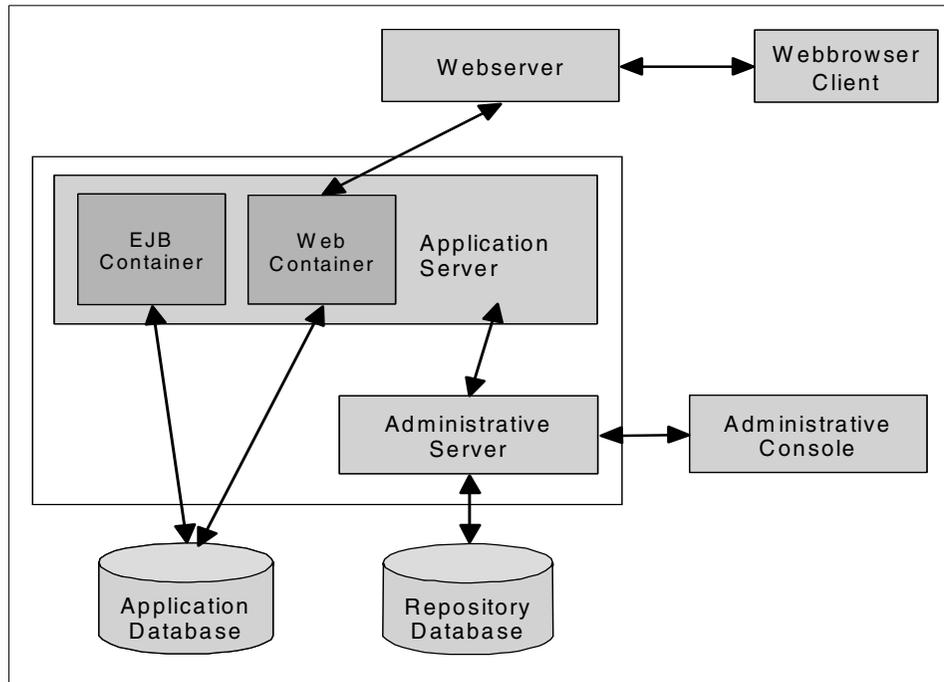


Figure 11-1 Components of WebSphere V4.0 Advanced Edition

In our case, we installed all components on one machine. A detailed list of the software components in our system is shown in Table 11-1.

Table 11-1 Software components

Software component	Version
Operating system	SuSE SLES 7 for S/390 and zSeries (31-bit)
Application server	WebSphere Application Server V4.02 Advanced Edition for Linux/390
Web server	IBM HTTP Server 1.3.19.3 for Linux/390
Database	Oracle9i Database Release 2 (9.2.0.1) Enterprise Edition for Linux/390

## 11.2 Installing the repository database

To run WebSphere with the repository database on another machine you have to install the Oracle client on the machine where your WebSphere is going to run. On the system we used for the WebSphere installation there was already a database server and a database installed as described in Chapter 3, “Installing Oracle9i Database on zSeries with Linux” on page 27. We only had to create a new database for the WebSphere Repository and configure the network services.

To create a new database we used the Oracle Database Configuration Assistant (DBCA). With this tool, we created a database called was40 and used the template for a general purpose database. Notice that the name for the database is not fixed to was40.

After the installation, we started the instance and made sure that the network functions of Oracle were running by checking the status of the listener and connecting from another machine.

### *Example 11-1 Status of the listener*

---

```
oracle@linux1:~ > lsnrctl status was40.itso.ibm.com

LSNRCTL for Linux: Version 9.2.0.1.0 - Production on 08-OCT-2002 13:03:12

Copyright (c) 1991, 2002, Oracle Corporation. All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=linux1)(PORT=1521))(CONNECT_DATA=(SER
VER=DEDICATED)(SERVICE_NAME=was40.itso.ibm.com)))
STATUS of the LISTENER
-----
Alias                LISTENER
Version              TNSLSNR for Linux: Version 9.2.0.1.0 - Production
Start Date           01-OCT-2002 12:53:24
Uptime               7 days 0 hr. 9 min. 48 sec
Trace Level          off
Security             OFF
SNMP                 OFF
Listener Parameter File /oracle1/dbserv/network/admin/listener.ora
Listener Log File    /oracle1/dbserv/network/log/listener.log
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=linux1.itso.ibm.com)(PORT=1521)))

Services Summary...
Service "PLSExtProc" has 1 instance(s).
  Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
```

```
Service "was40.itso.ibm.com" has 2 instance(s).
  Instance "was40", status UNKNOWN, has 1 handler(s) for this service...
  Instance "was40", status READY, has 1 handler(s) for this service...
Service "was40XDB.itso.ibm.com" has 1 instance(s).
  Instance "was40", status READY, has 1 handler(s) for this service...
The command completed successfully
```

---

## 11.3 Configuration of the repository database

WebSphere requires two users to create tables in the repository database. To create them, perform the following steps:

1. Log into the system as user oracle:

```
oracle@linux1:~ > id
uid=59(oracle) gid=54(oinstall) groups=54(oinstall),55(dba)
```

2. Set your ORACLE\_SID to was40, or whatever name you have chosen for your database:

```
oracle@linux1:~ > export ORACLE_SID=was40
```

3. Connect to the database:

```
oracle@linux1:~ > sqlplus
```

```
SQL*Plus: Release 9.2.0.1.0 - Production on Tue Oct 8 13:33:01 2002
```

```
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.
```

```
Enter user-name: / as sysdba
```

```
Connected to:
```

```
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production
```

4. If not already done, start up the database:

```
SQL> STARTUP
```

5. Create the database users EJSADMIN and EJB. If you like to grant user EJSADMIN dba authority follow step a. Otherwise, go to step b:

a.

```
SQL> CREATE USER EJSADMIN IDENTIFIED BY <anypassword>;
SQL> GRANT connect, resource, dba TO EJSADMIN;
SQL> CREATE USER EJB IDENTIFIED BY EJB;
SQL> GRANT connect, resource TO EJB;
SQL> QUIT
```

b.

```
SQL> CREATE USER EJSADMIN IDENTIFIED BY <anypassword> QUOTA 100M ON
SYSTEM;
SQL> GRANT connect, resource TO EJSADMIN;
SQL> CREATE USER EJB IDENTIFIED BY EJB QUOTA 100M ON USERS;
SQL> GRANT connect, resource TO EJB;
SQL> QUIT
```

#### 6. Test the access to your database:

```
oracle@linux1:~ > sqlplus EJSADMIN/<anypassword>
SQL*Plus: Release 9.2.0.1.0 - Production on Tue Oct 8 13:56:59 2002
```

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

```
Connected to:
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production
```

```
SQL>QUIT
```

## 11.4 Installing IBM HTTP server

You can use different Web servers together with WebSphere. On Linux for zSeries you can choose between the Apache Webserver and the IBM Web server. Before you install one of them, see the prerequisites to find out which versions are supported in conjunction with WebSphere. The prerequisites can be found on the following Web site:

<http://www-3.ibm.com/software/webservers/appserv/doc/latest/prereq.html>

After registration you can download the installation files for the IBM HTTP Server from this Web site:

<http://www6.software.ibm.com/dl/websphere/http-p>

Before you proceed with the installation, you should check, if there is already a Web server running on your system. Stop it if there is one. The standard installation of SuSE SLES 7 includes the Apache Webserver. To stop it and also prevent it from being started after a reboot, you have to log into the system as root:

#### 1. Stop the Web server:

```
linux1:~ # rcapache stop
Shutting down httpd
done
```

2. Disable the Web server after the next reboot by editing the file `/etc/rc.config`. Change the line:

```
START_HTTPD=yes  
to  
START_HTTPD=no
```

Now you can uncompress and extract the downloaded file and install the RPMs for the Web server. The IBM HTTP server is installed into the directory `/opt/IBMHTTPServer`, and its configuration files are stored in the directory `/opt/IBMHTTPServer/conf`.

At the beginning of the installation we got the following error message:

```
linux1:~/IHS-1.3.19.3 # rpm -ivh IBM_HTTP_Server-1.3.19-3.s390.rpm  
error: failed dependencies:  
    libstdc++-libc6.1-2.so.3 is needed by IBM_HTTP_Server-1.3.19-3
```

We tried to solve the problem by creating a symbolic link in the directory `/usr/lib`:

```
linux1:/usr/lib # cd /usr/lib  
linux1:/usr/lib # ln -s libstdc++-3-libc6.2-2-2.10.0.so  
libstdc++-libc6.1-2.so.3  
linux1:/usr/lib # ldconfig
```

As it did not solve the problem, we finally installed the package with the `nodeps`-option:

```
linux1:~/IHS-1.3.19.3 # rpm -ivh --nodeps IBM_HTTP_Server-1.3.19-3.s390.rpm
```

To test the Web server, we put a simple HTML file into the root directory of the Web server. You can find out and change the root directory of the Web server by editing the file `/opt/IBMHTTPServer/conf/httpd.conf` and searching for the line beginning with `DocumentRoot`.

## 11.5 Installing WebSphere

To start the installation of WebSphere mount your installation medium and run the script `install.sh`. Make sure that you have set your `DISPLAY` variable correctly and that an X-server is running on your machine. If everything is set properly, you should get a welcome screen. The next steps are:

1. If you do not have an older version of WebSphere on your machine that you want to migrate, choose the **Do not migrate** option and click **Next**.
2. Choose the **custom installation** and click **Next**.

3. On the next screen you can select the components you want to install. Figure 11-2 shows the sample of our test installation. Click **Next**.

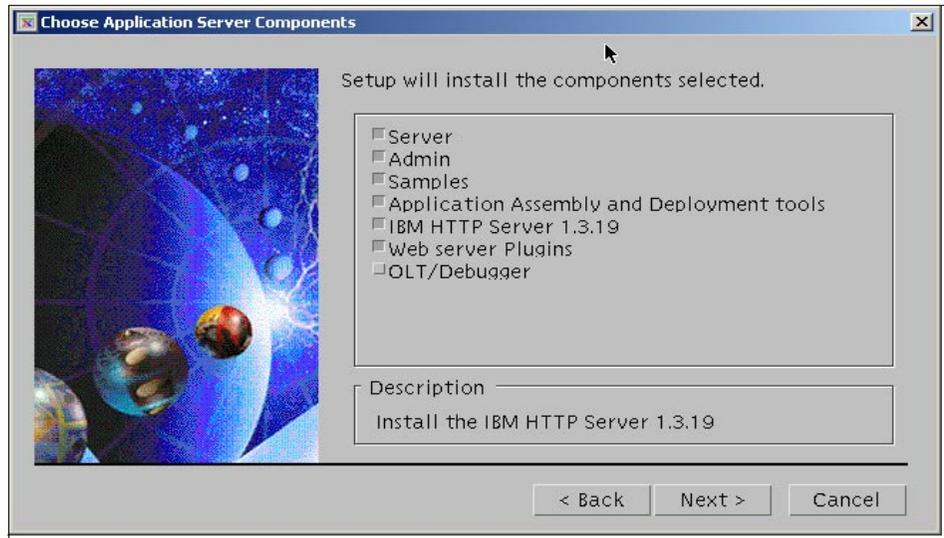


Figure 11-2 Installed components

4. Select the **IBM HTTP server plug-ins** and click **Next**.
5. Figure 11-3 on page 181 shows the next screen of the installer and the values for our installation. There you have to fill out the information about your repository database:
  - a. First you have to choose the database type, which is **Oracle**.
  - b. Activate the button for a **remote database**.
  - c. Enter the database name of the repository database, the home directory of your Oracle installation, the name of your server, and the port number of your listener. Refer to Chapter 11.2, "Installing the repository database" on page 176.
  - d. Enter the database user ID and the password as you created them in Chapter 11.3, "Configuration of the repository database" on page 177.
  - e. Click **Next**.

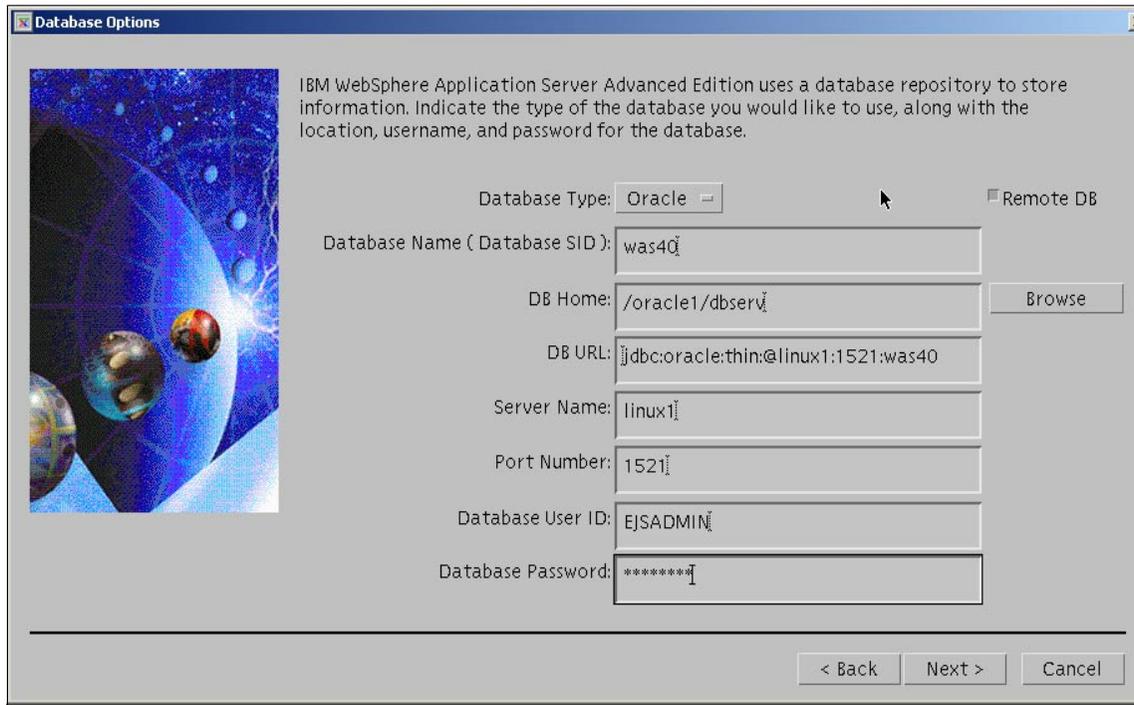


Figure 11-3 Information about the repository database

6. Enter the path of the destination directory of your application server and your Web server. Our installation directories:

Application server/opt/WebSphere/AppServer  
 Webserver/opt/IBMHTTPServer

Click **Next**.

7. Check your installation settings and click **Install**.
8. Enter the path to the IBM HTTP server configuration file (/opt/IBMHTTPServer/conf/httpd.conf), then click **Next**.
9. Click **Finish** to end your installation.

## 11.6 Running WebSphere as non-root user

For security considerations you should not run WebSphere as user root in a production environment. The first step to changing it, is to create as root a user and a group for WebSphere. Then change the file ownership of all files under the WebSphere installation directory to this user and group.

```
linux1:/ # groupadd websphere
linux1:/ # useradd -d /opt/WebSphere/AppServer -g websphere websphere
linux1:/ # cd /opt/WebSphere
linux1:/ # chown -R websphere:websphere *
```

Furthermore, you have to change the port number on which the application server is listening. The default port is 900. However, all ports below 1024 are restricted to administrative or super user use only. To change the port, first figure out which port above 1024 you can use. Have a look in the file `/etc/services` to find a port that is not reserved. In our configuration we used port number 4700. Now edit the file `/opt/WebSphere/AppServer/bin/admin.config` and add the line:

```
com.ibm.ejs.sm.adminServer.bootstrapPort=4700
```

After saving the file, you are able to start the application server as user `WebSphere`.

## 11.7 Starting the application server

Before you can start the application server, you have to start the administrative server. Change to the `WebSphere` installation directory and run the script `startupServer.sh`. Make sure that you are logged in as the user you created for running `WebSphere`:

```
websphere@linux1:/ > id
uid=501(websphere) gid=101(websphere) groups=101(websphere)
websphere@linux1:/ > cd /opt/WebSphere/AppServer/bin
websphere@linux1:/opt/WebSphere/AppServer/bin > ./startupServer.sh &
```

To determine if the administrative server is coming up, examine the logfile named `tracefile`, that is stored in the directory `logs` under the `WebSphere` installation directory. When the message `adminServer open for e-business` appears, the administrative server has started successfully:

```
websphere@linux1:/opt/WebSphere/AppServer/logs > tail -f tracefile
[10/7/02 9:13:25:868 EDT] 4d679a94 ResourceBinde I WSVR0049I: Binding
SM_DATASOURCE as jdbc/SM_Datasource
[10/7/02 9:13:30:400 EDT] 4d679a94 EJBEngine I WSVR0037I: Starting EJB
jar: Name Service
[10/7/02 9:13:39:678 EDT] 4d679a94 EJBEngine I WSVR0037I: Starting EJB
jar: Secure CosNaming
[10/7/02 9:13:41:238 EDT] 4d679a94 EJBEngine I WSVR0037I: Starting EJB
jar: Repository
[10/7/02 9:13:56:583 EDT] 4d679a94 EJBEngine I WSVR0037I: Starting EJB
jar: Tasks
[10/7/02 9:14:05:607 EDT] 4d679a94 Server A WSVR0023I: Server
__adminServer open for e-business
```

Now you can connect your administrative console to the administrative server and start the application server. To start the administrative console, run the script `adminclient.sh` from your WebSphere installation directory. As we have changed the default port for WebSphere, you need to start the script with the parameters of the host address and the port number. Again, make sure that an X-server is running on your machine and that the `DISPLAY` variable is set:

```
websphere@linux1:/ > cd /opt/WebSphere/AppServer/bin
websphere@linux1:/opt/WebSphere/AppServer/bin > ./adminclient.sh localhost
4700
```

Once you have entered the administrative console, you can start your application server by marking it and clicking the **Start** button in the toolbar.

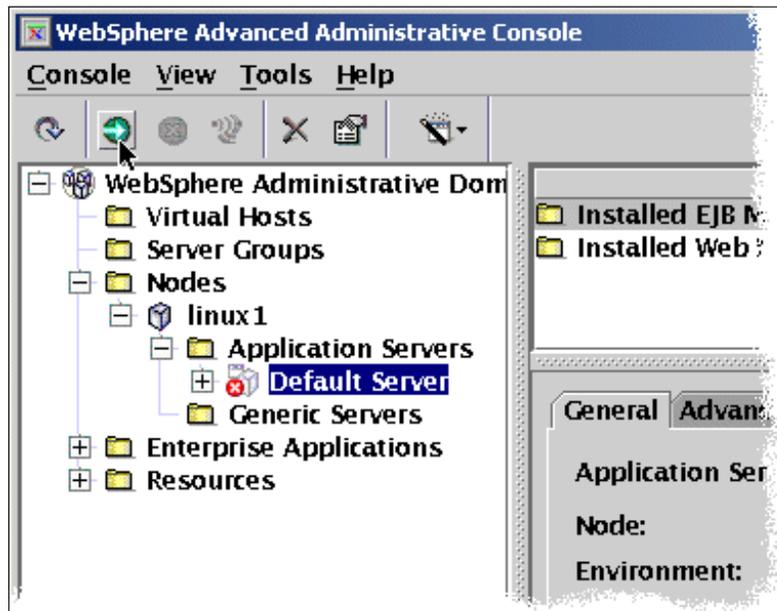


Figure 11-4 Starting the default application server

## 11.8 Testing the installation

There is a set of sample applications which you can use to test your configuration. Before you try to implement your own applications, you should first check if the application server is running these sample applications.

Go to your Web browser and enter the following URL where host address has to be substituted with the address of your Web server:

```
http://<hostaddress>/webapp/examples/
```

You should get a list of examples and if you follow the links, the sample applications are run.

To test the jdbc driver and the connection to a remote Oracle database, you can run another sample application. Before you can start it, you have to install a database schema on the remote database and configure the connection to it. You can find a detailed description of how to do this, when you enter this URL:

```
http://<hostaddress>/WSsamples
```

In our tests, we used the pet store sample and were able to run the application with a remote Oracle database, but we had to change the SQL script that creates the database user and tables. At the beginning of the script (at the point of creating the user) we had to remove the double quotes around the username. If you follow the other steps of the instructions, you should be able to run the sample application.



**A**

## Frequently asked questions about Oracle9*i* for zLinux

Q1. What, specifically, did Oracle announce on March 18, 2002?

A1. Oracle announced that they will develop and support a 31-bit version of Oracle9*i* database for Linux on zSeries (both Enterprise and Standard Editions). Since November 2001, a developer's release of this product has been available for download from the Oracle Technical Network. The Standard Edition is available for systems (including LPARS) that have four or less processors.

Q2. What are the official names of the Oracle products for Linux/390?

A2. The official names are:

- ▶ Oracle9*i* Enterprise Edition Release 1 (9.0.1) for Linux/390 (long name for Oracle's 31-bit version of the database server product)
- ▶ Oracle9*i* Database for Linux/390 (short name for Oracle's 31-bit version of the database server product)
- ▶ Oracle9*i* Enterprise Edition Release 1 (9.0.1) for z/Linux (long name for Oracle's 64-bit version of the database server product)
- ▶ Oracle9*i* Database for z/Linux (short name for Oracle's 64-bit version of the database server product)

Q3. What is the scope of Oracle9*i* database product for Linux/390?

A3. Oracle9i Enterprise Edition Release 2 for Linux/390 is available with all features including Real Application Clusters (RAC), with the additional following exceptions to those product-specific restrictions listed in the Oracle9i Installation Guide Release 2 (9.2.0.1.0) for UNIX systems:

The following precompilers are not supported:

- Pro\*COBOL (planned for 1Q/03)
- Pro\*FORTRAN
- SQL Module f or A da

The following Oracle Advanced Security protocols are not supported:

- Radius challenge response authentication
- CyberSafe
- DCE Integration
- Entrust
- nCipher

Generic Connectivity is not supported.

The INSO\_FILTER is not available for Oracle Text.

Q4. There is a 64-bit version of the Developer Release Oracle9i Release 1 available for download. Has Oracle announced a general availability date for this version?

A4. No, the intent is to announce a 64-bit version product at a later date.

Q5. Will Oracle9i/AS be announced for Linux on zSeries?

A5. The feasibility of offering Oracle9i/AS will be investigated.

Q6. Can Oracle's e-business Suite be deployed in mixed mode using the Oracle9i database on Linux/390?

A6. Not at this time.

Q7. On what other Linux platforms is Oracle9i supported?

A7. Oracle products run on Linux on Intel platforms.

Q8. Is z/VM required?

A8. Oracle9i for Linux/390 has been successfully tested in native LPARs and under z/VM. We recommend deployments to be under VM, if you will be moving workload from many small Oracle instances to Linux on zSeries. If you are moving one or two applications to Oracle on zLinux, and do not have in-house VM skills, expect deployments may be in native LPARs.

Q9. Are there presently customers running Oracle9i for Linux/390 database?

A9. There have been over 1000 downloads of the developer's release of the code. Several customers were part of an early test program. Contact your local Oracle representative for more information. The OTN Web site can be found at: <http://www.oracle.com> then choose **Oracle Technology Network** on the left side of the panel. Linux is the faster growing operating system; therefore, we expect fast growth on this platform.

Q10 Has performance been evaluated?

A10. Preliminary testing has shown performance to be satisfactory. Performance of any application is always dependant on many factors including the customer's specific implementation, the available hardware configuration, operating system, compilers, capacity, etc. Oracle9i , Release 2 performs best when resources are not over committed.

Q11. How will Oracle price this product?

A11. Contact your local Oracle representative or go to <http://store.oracle.com> Currently, Oracle pricing is based on the number of engines available for Oracle work.

Q12.What Linux distributions are supported?

A12. The plan is for Oracle9i, Release 2 for Linux/390 has been developed on SuSE and is being certified on Red Hat.

Q13. What documentation is available?

A13. There are specific release notes available on OTN. Oracle generic documentation for Linux Intel is applicable.

Q14, Why would customers want to run Oracle9i on Linux/390?

A14. There are many small servers running applications on Linux today. The zSeries offers unique benefits to consolidate Linux servers under VM to allow for full exploitation of capacity, and for easy of use of systems management. This should provide benefits to reduce the total cost of computing. It will also allow customers to easily expand their use of the Oracle9i database on Linux/390.

Q15.What is the sizing methodology?

A15. This is dependent on the applications used and the customer's individual implementation. IBMers can get assistance from their IBM team.

Q16. Why is IBM supporting Oracle9i on S/390 and zSeries?

A16. IBM supports solutions its customers require.

Q17. Can Oracle9i for Linux/390 run on CMOS hardware?

A17. Oracle9i 31-bit will run on any hardware that can host Linux/390.

Q18. Where should customers, worldwide, call for more information on Oracle9i for Linux/390?

A18. Customers should contact their local IBM or Oracle sales representative, or visit:

<http://www.ibm.com/eserver> and <http://www.oracle.com>

Oracle and IBM sales representatives can contact the IBM/Oracle International Competency Center at:

<http://www.ibmoracle@us.ibm.com>

Q19. Does Oracle Enterprise Manager run with Oracle 9i on zLinux?

A19. OEM can run on zLinux. You cannot run the OEM Management Server on zLinux because it uses a back level of the JDK that is not available. OMS can be run on another platform. You can access and manage your Oracle9i instances on zSeries from OMS on the other platform.



## Tools for VM performance data collection and analysis

This appendix is intended for those who are not familiar with z/VM, but are beginning to use it for the first time to run Linux guests. It provides an overview of the tools available (from IBM) in the z/VM environment to measure and analyze performance data. Some of these facilities are provided by z/VM itself, and some are additional optional priced features or products. Except where noted below, all of these tools provide information about resource utilization or contention at the virtual machine level. They do not provide information about utilization or consumption within the Linux virtual machines.

## z/VM Control Program (CP)

The z/VM Control Program (CP) produces performance and resource consumption information in two different ways:

- ▶ CP Indicate command
- ▶ Accounting records

### B.0.1 The CP INDICATE command

The INDICATE command can be used by system resource operators, system programmers, system analysts, and general users to display on the console the use of and contention for system resources. It is often used to get a quick, immediate look at system or user resource consumption. If a performance problem is suspected, this is often the quickest way to identify the characteristics of the problem. There are several variations of the INDICATE command that are useful in gathering performance data:

1. INDICATE USER command. General users can enter the INDICATE USER command to obtain system resource usage statistics about their own virtual machines. A system analyst can enter the INDICATE USER command to obtain statistics for any virtual machine. CP displays a set of statistics for each of the virtual machine's virtual processors that provide a snapshot of the status of the virtual machine at the time the command is issued. These statistics include values such as:
  - a. The number of pages that the user has resident in real storage.
  - b. The most recent estimate of the virtual machine's working set size.
  - c. The total number of paging operations for the virtual machine since it was logged on.
  - d. The current number of blocks allocated in expanded storage for this virtual machine.
  - e. The virtual CPU time, and virtual CPU time plus simulation time for each virtual processor.
  - f. The total number of non spooled I/O requests since the virtual machine was logged on.
- g. INDICATE LOAD command. The system analyst can enter an INDICATE LOAD command to obtain system information that indicates current contention for system resources. A general user can receive a subset of the information. Statistics displayed for the system analyst include:
  - i. Processor utilization
  - ii. Storage utilization (central & expanded) & paging activity

- iii. Minidisk cache activity
- iv. The number of users in the various z/VM scheduler queues
- v. User expansion factors, which indicate the delay in response time because of contention for resources.
- h. Other INDICATE COMMAND options. Other variations of the INDICATE COMMAND can be used to gather more detailed information about the usage of or contention for system resources. Some of these are INDICATE QUEUES, INDICATE IO and INDICATE PAGING. The INDICATE COMMAND is described in detail in SC24-6008-01, z/VM Command and Utility Reference.

## B.0.2 Accounting records

Accounting records. Virtual machine accounting records can be produced that contain information about the resources used by individual virtual machines. They can be created at any time but are automatically created when a virtual machine logs off. These records contain information about resource consumption since the virtual machine logged on, or since the last accounting records were cut. Since these statistics are usually accumulated over larger time intervals, they are not often useful for analyzing performance problems, but can be useful in capacity planning activities or for accounting charge-back purposes. The accounting records that may be of the most use are the “virtual machine Resource Usage” records (type 01). The information provided by these records includes the userid, cpu time used, number of page reads/writes and number of I/O operations. z/VM 4.3 adds some new accounting records for measuring the amount of data flowing across virtual network connections such as guest LANs and IUCV or virtual channel-to-channel connections.

These accounting records are created by the z/VM Control Program (CP) and passed to designated virtual machines via the CP system service \*ACCOUNT. It is the responsibility of the receiving virtual machine to receive the records and store them on disk. Up to two virtual machines can be designated in the SYSTEM CONFIG file to receive accounting records and these virtual machines will be automatically logged on when z/VM is IPLed. Sample directory entries and PROFILE EXECs for these machines are provided so that it is easy to set them up to automatically begin collecting accounting records each time the system is IPLed. IBM does not provide any programs for processing these records but the record layouts and the detailed instructions for collecting the records are described in SC24-5995-01, z/VM 4.2 Planning and Administration.

### B.0.3 CP Monitor Facility

The CP Monitor Facility. The primary source of performance data in a z/VM system is the CP Monitor Facility. The data provided by the CP Monitor is useful for a variety of activities, from in-depth performance problem analysis to long term capacity planning and trend analysis. The data is usually processed by an external data reduction program to produce statistics to give an understanding of system operation or help analyze the use of, and contention for, major system resources. These resources include processors, storage, I/O devices, and the paging subsystem. The installation can control the amount and nature of the data collected. The general process of collecting monitor data is:

- a. The CP privileged command, MONITOR, is used to start/stop the collection of data and to specify the type, amount, and nature of data to be collected.
- b. The monitor collects performance data during CP operation and stores it, in the form of monitor records, in a saved segment.
- c. An IBM supplied application program, MONWRITE, running in a CMS virtual machine connects to the CP \*MONITOR System Service to establish a data link with CP. MONWRITE retrieves monitor records from the saved segment and stores them on tape or in a CMS file on disk. An application program can then later read the records from the file and perform data reduction on the performance data found in the records.

The CP Monitor collects and reports two types of data: event data and sample data:

- ▶ Event data is collected and reported each time a designated system event occurs. The data reported represents the status of the system at the time the event occurred.
- ▶ Sample data is collected and reported at the end of each designated time interval. The varieties of sample data are:
  - Single-sample data, which is collected once during the time interval, and only at the end of that time interval. Some of this data represents the status of the system at the time the data was collected; other data is accumulations of counters, states, or elapsed times collected at the end of each time interval.
  - High-frequency sample data, which is collected at a rate higher than it is reported on. The data is reported along with single-sample data. At each high-frequency sampling time, the data collected is added to its corresponding counters; the data reported is an accumulation of counts or states that had been collected from the time high-frequency sampling started.

Through the MONITOR command, you can select which or both of the types of data to collect. You can also control the time interval for single-sampling and the rate for high-frequency sampling. Instructions for gathering performance data using the CP Monitor, including defining the Monitor Saved Segment, setting up the retrieval virtual machine, specifying the types of data to be collected and starting/stopping the monitor data collection can be found in *z/VM Performance*, SC24-5999-01.

## B.1 Optional programs

IBM also provides several optional programs that are either priced features of z/VM or separate products to assist with performance management in the VM environment:

### B.1.1 VM Performance Reporting Facility (VMPRF).

VMPRF, reads and processes the data produced by MONWRITE, mentioned above. It produces performance data reports and history files from the monitor data. VMPRF detects and diagnoses performance problems, analyzes system performance, and provides printed reports and trend data showing performance and usage of the z/VM system. VMPRF simplifies performance analysis and resource management of z/VM. Reports and history files produced by VMPRF include information on:

- System resource utilization, transaction response time, and throughput
- Resource utilization by user ID
- DASD activity and channel utilization
- VMPRF also provides:

User information by user class - Information reported by user ID can also be reported by classes of user. The mapping of user IDs to user classes is specified in a CMS file.

Interim reports - Interim reports reflect performance during smaller periods of the usual reporting period. For example, if an 8-hour shift is the usual reporting period, interim reports might report on each hour of that period.

History files - VMPRF keeps key performance measures on CMS disk files, in an easy-to-use format, for further processing. Two types of history files are provided:

SUMMARY files for analyzing

TREND files for long term trend data

Beginning with z/VM Version 4, VMPRF is now distributed as an optional, priced feature of z/VM.

However the z/VM 4.3 announcement letter (202-104) on April 30, 2002, contained a statement of direction to withdraw VMPRF from marketing in a future release of z/VM. The intended replacement product will be based on FCON/ESA, which is described later in this paper. For more information see: <http://www.vm.ibm.com/related/prf/>

## B.1.2 VM Real Time Monitor (RTM)

- The VM Real Time Monitor provides an immediate view of the current system performance. RTM can be used for short term monitoring, analysis and problem solving. It has easy to use commands to display performance data and can produce a performance log that shows the values of key performance indicators at time intervals throughout the measurement period. RTM does not get its information from the CP Monitor, but instead extracts its input directly from CP internal control blocks so there is no need to have the CP monitor running to use RTM. RTM runs in a CMS virtual machine and other users on the system can send commands to the RTM virtual machine to request displays of performance data. One can request various types of displays from RTM, from very high level overall system information to very specific information about the performance of individual subsystems, users, or resources. Some examples of the RTM displays are:

The GENERAL display is a good starting point for diagnosing performance problems. It shows CPU, storage and paging statistics for the most active users, I/O rates for the most active devices and overall CPU and storage utilization statistics for the entire system.

The USER display shows CPU, storage and paging resource usage for all users or individual users.

The DEVICE display contains information about resource utilization in the I/O subsystem. This screen provides data kept by CP, as well as data from the hardware Channel Measurement Facility.

The CPU display contains interval and cumulative data for each individual processor currently varied online in the configuration.

The SYSDASD display contains paging and spooling related information for all system DASD that have spool or page space allocated.

Like VMPRF, RTM is distributed as an optional priced feature of z/VM Version 4. And the statement of direction mentioned above also states that RTM will be withdrawn from marketing in a future release of z/VM. The intended replacement product for RTM will be the same future product based on FCON/ESA, which is described later in this paper. More information on RTM can be found at: <http://www.vm.ibm.com/related/rtm/>

### B.1.3 RMF PM

RMF PM is provided as part of an IBM technology study. It was originally developed for OS/390 RMF but has been extended to support Linux. It consists of a data collector program that runs on Linux and a Java display client that can run on other platforms. Collector programs are provided for the Intel Linux platform and for Linux for S/390 (31-bit) and Linux for zSeries (64-bit). Java display clients are provided for both Windows and Linux. The client can make performance data accessible through XML over HTTP so it can be utilized by other applications. Using the RMF PM client, it is possible to:

- ▶ Generate graphical trend reports
- ▶ Store the data in spreadsheet format (Lotus 1-2-3, StarOffice, MS Excel, etc.)
- ▶ Persistently store performance analysis
- ▶ Gather historical performance data
- ▶ Filter the performance data
- ▶ Mix OS/390 and Linux performance data in one screen
- ▶ Use a graphical user client to access performance data in a flexible and highly configurable way

More information and the data collector and display clients can be found at: <http://www.s390.ibm.com/rmf/rmfhtmls/pmweb/pmlin.htm>

### B.1.4 FCON/ESA

This is a product from IBM Switzerland that is another real-time performance monitor. FCON/ESA provides performance monitoring capabilities in full screen mode. It can provide an immediate view of system performance or post process its own history files or CP Monitor data. Threshold monitoring, user loop detection and the ability to monitor remote systems is also provided. Data can be provided in HTML format and viewed from a web browser. Recently FCON/ESA has been enhanced to process data provided by the RMF PM data collector mentioned above. This allows it to combine and display both VM and Linux performance data. Currently, this product can only be ordered directly from the author, but as indicated above it is IBM's intent to replace both VMPRF and RTM

with a new product based on FCON/ESA. Introductory information (although quite old) can be found at <http://vm.ibm.com/perf/FCXABSTR.PDF>. More recent information on FCON/ESA and RMF PM and a complete review of Linux performance tools can be found in Oliver Benke's presentation, Monitoring & Understanding Performance on Linux for S/390 & zSeries, given at Share in Nashville (session 5542). A copy of the presentation can be found at the Share Web site at:

<http://www.share.org/proceedings/SH98/share.htm>



# C

## Examples of setting up the xWindows Interface

This appendix shows examples of how we set up connectivity to run the OUI with the xWindows Interface using:

- ▶ A Linux Intel system
- ▶ VNC software
- ▶ Exceed software

## Using a Linux Intel system as a client

The steps to use a Linux Intel system are:

1. Logon to Linux Client as root.
2. Issue the command **xhost +**  
the + gives everyone access to the system.
3. Issue the command **startx**  
Only necessary if xWindows is not initialized.
4. Open another window for the Telnet session.
5. Telnet to the zSeries Linux system, logon as oracle, issue the command **export DISPLAY=hostname:0.0**  
where hostname is the hostname or IP address of the Linux Intel Client and 0 is the default console.
6. Test by issuing the command **xclock &**

Now you are ready to start the Oracle Universal Installer.

## Using VNC client and server

We have the VNC server code installed on a Linux Intel system that was located on the same LAN as the zSeries.

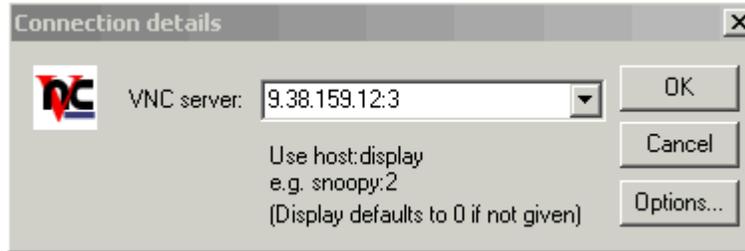
The IP addresses we used were:

- ▶ VNC server: 9.38.159.12
  - Port 3 was enabled for our workstation.
- ▶ Workstation Client with VNC viewer: 9.38.159.141
- ▶ ZSeries with Linux system: 9.38.159.11

You can also install the VNC server on the Linux/zSeries machine directly. The rpm package is included in the SLES7 distribution for zSeries. You can install the VNC server with YaST or manually install the package that is located in the directory /suse/xap2 of the SuSE distribution. After the installation, simply run the command **vncserver** as the oracle user. Enter a password to connect to your vncserver and start the vncviewer from your workstation as described in our the next step.

From the workstation:

1. Invoke the VNC viewer.
2. Enter IP and port number of the VNC server.



3. Enter the password.



4. Click and choose **Xterm option**.
5. In the window, Telnet to the Linux zSeries system  
telnet 9.38.159.11
6. Enter the user ID and password.
7. Issue the command **xhost +**
8. Issue the command **su - oracle**
9. Issue the command **cd /ora92/Disk1**
10. Issue the command **EXPORT DISPLAY=9.38.159.12:3**

You can check by using the command `xclock &`

If this displays a window with a clock, you are ready to start the OUI by entering

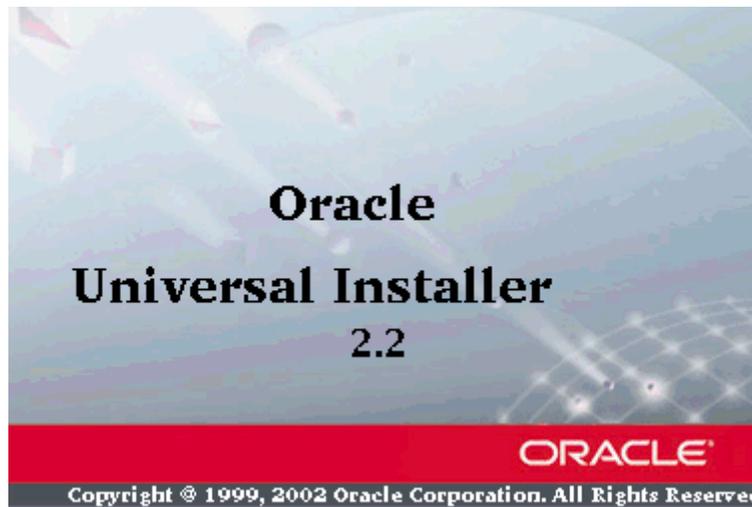
11. `./runInstaller`

## Using Exceed with Telnet

The steps to enable your Linux guest to export the xWindows display to your workstation using Exceed are:

1. Go to Exceed, host access, Telnet.
2. Connect to the zSeries Linux system.  
Telnet 9.12.159.11
3. Enter the user ID and password.

4. Go to Exceed, xConfig.
  - Click **Configuration**.
  - Check that it is in passive mode.
  - Click the screen definition.
  - Check that window mode is multiple.
5. Go to Exceed, start Exceed to have it running as a background process.
6. Return to the Telnet session.
7. Enter `ioczm10s:~ # su - oracle`
8. Enter:  
`oracle@ioczm10s:~ > export DISPLAY=9.38.157.215:0`  
where hostname is the name of your workstation or the IP address of your workstation.
9. Enter:  
`oracle@ioczm10s:~ > xhost +`
10. Access control disabled, clients can connect from any host.
11. xhost: must be on local machine to enable or disable access control.
12. Change directories to where you have staged the install CDs:  
`cd /ora92`
13. Change directory to the first disk:  
`cd Disk1`
14. Run the Oracle Universal Installer:  
`./runInstaller`  
`oracle@ioczm10s:/ora92/Disk1 > Initializing Java virtual machine from ../../sta`  
This screen will appear for a few seconds and then the Welcome screen will appear.





# D

## **Linux Performance Data Collection and Analysis Tools**

This appendix is intended for those who are not familiar with Linux on zSeries, but are beginning to use it for the first time to run Linux guests. The described tools `vmstat`, `xosview`, and `top` are part of the standard SuSE installation, and need no further installation or configuration. Most of the tools fetch their information from the `proc` filesystem. You can read the information in the `proc` filesystem by using the linux command `cat`. But, generally the output of the following tools are easier to read.

# VMSTAT

Vmstat is a command line tool. It dumps system information to the standard output. You invoke it by typing **vmstat** on the command line. With two parameters you can control vmstat. The first is the delay between updates, and the second is the count of updates. In Example D-1 vmstat is run five times at intervals of ten seconds.

*Example: D-1 Output of the vmstat command*

---

```
linux1:/etc # vmstat 10 5
```

procs				memory				swap		io		system			cpu	
r	b	w	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	
1	0	0	65072	4468	13004	421048	0	0	1	5	0	8	0	1	29	
0	0	0	65072	4468	13004	421048	0	0	0	8	0	10	0	1	99	
0	0	0	65072	4468	13004	421048	0	0	0	10	0	11	0	0	100	
0	0	0	65072	4436	13004	421080	0	0	0	10	0	11	0	0	100	
0	0	0	65072	4424	13004	421092	0	0	0	14	0	11	0	0	100	

---

# XOSVIEW

Xosview is a small X based system monitor. It displays some of the current system values, like CPU and memory usage and network traffic. Xosview is installed with the SuSE standard installation. It is started by simply typing the command `xosview &`. Like all other X based programs you need a running X server, and you have to set the DISPLAY variable before you start xosview. This tool is not meant to be a detailed analyzing tool. But for performance monitoring during a short period, such as an import operation, it is very practical.

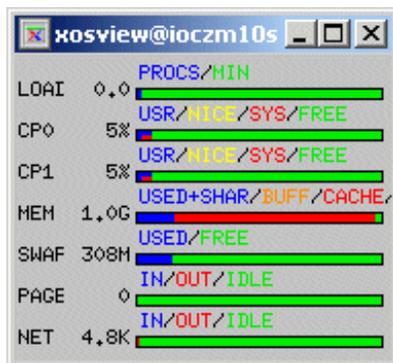


Figure 11-5 Xosview Window

# TOP

Top displays the running processes in real time and how many resources they use. Its display is limited to the number of lines you have in your terminal. When you start top in the interactive mode, what is the default, there are a variety of key functions to manipulate the output. If you press H, you get a short summary of which functions are available. The most common used are:

**space** update the display  
**c** switch between command name and command line display  
**u** show only specific user  
**A** sort by age  
**P** sort by CPU usage

*Example: D-2 Sample output of the top command*

---

4:09pm up 14 days, 2:19, 2 users, load average: 0.00, 0.00, 0.00  
52 processes: 50 sleeping, 2 running, 0 zombie, 0 stopped  
CPU states: 0.0% user, 2.3% system, 0.0% nice, 97.6% idle  
Mem: 511656K av, 507932K used, 3724K free, OK shrd, 13004K buff  
Swap: 143896K av, 65072K used, 78824K free 421648K cached

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	COMMAND
12473	root	18	0	1048	1048	856	R	0.5	0.2	0:00	top
1	root	8	0	80	64	64	S	0.0	0.0	0:01	init
2	root	9	0	0	0	0	SW	0.0	0.0	0:00	kmcheck
3	root	9	0	0	0	0	SW	0.0	0.0	0:00	keventd
4	root	19	19	0	0	0	SWN	0.0	0.0	0:00	ksoftirqd_CPU0
5	root	9	0	0	0	0	SW	0.0	0.0	0:07	kswapd
6	root	9	0	0	0	0	SW	0.0	0.0	0:00	kreclaimd
7	root	9	0	0	0	0	SW	0.0	0.0	0:00	bdfush
8	root	9	0	0	0	0	SW	0.0	0.0	0:00	kupdated
98	root	9	0	0	0	0	SW	0.0	0.0	0:00	qethsoftd0000
126	bin	9	0	84	4	4	S	0.0	0.0	0:00	portmap
138	root	9	0	168	4	4	S	0.0	0.0	0:00	sshd
147	root	9	0	284	240	240	S	0.0	0.0	0:00	syslogd
152	root	9	0	636	216	216	S	0.0	0.0	0:00	klogd
165	at	9	0	172	116	116	S	0.0	0.0	0:00	atd
207	root	9	0	84	4	4	S	0.0	0.0	0:00	lpd
215	root	9	0	96	4	4	S	0.0	0.0	0:00	rpc.statd

---

## IOSTAT & SAR

iostat and sar are not part of the standard SuSE installation. They belong to the sysstat package. Both are command line tools. iostat reports input/output statistics for devices.

*Example: D-3 Sample iostat output*

---

```
ioczm10s:~ # iostat
Linux 2.4.7-SuSE-SMP (ioczm10s)          09/30/02

avg-cpu:  %user   %nice    %sys  %idle
           6.24    0.01    2.19  91.56

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
dev94-0            28.39      243.19       430.91  206946016  366688088
```

---

Sar offers history information about the system activity. To automate the data collection you have to run sar scheduled e.g. by the cron daemon. If you run sar without a parameter, it shows you the standard system activity daily data file as you can see in Example D-4.

*Example: D-4 Sample sar output*

---

```
ioczm10s:~ # sar
Linux 2.4.7-SuSE-SMP (ioczm10s)          09/30/02
00:00:00      CPU    %user   %nice  %system  %idle
00:10:00      all    0.51    0.00   0.62    98.87
00:20:00      all    0.35    1.01   1.60    97.04
00:30:00      all    0.32    0.00   0.37    99.30
...
07:30:00      all    0.36    0.00   0.43    99.22
07:40:00      all    0.37    0.00   0.34    99.28
Average:      all    0.41    0.02   0.62    98.95
```

---

## **BMC PATROL**

BMC Patrol is a commercial tool. Its function range goes far across the standard Linux tools. It does not only show the current system state, but also provides automated event notification.

# RMF PM with support for Linux enterprise server

RMF PM is a client server application that collects performance data on the server site. You can view the gathered information and generated reports either with a client software. Or, you use the Web interface of the tool, and a Web browser as the client. The required software can be downloaded from the following Web site:

<http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/pmweb/pmlin.htm>

The client software is available for Linux and Windows platforms. For further prerequisites and installation instructions see the Web site.

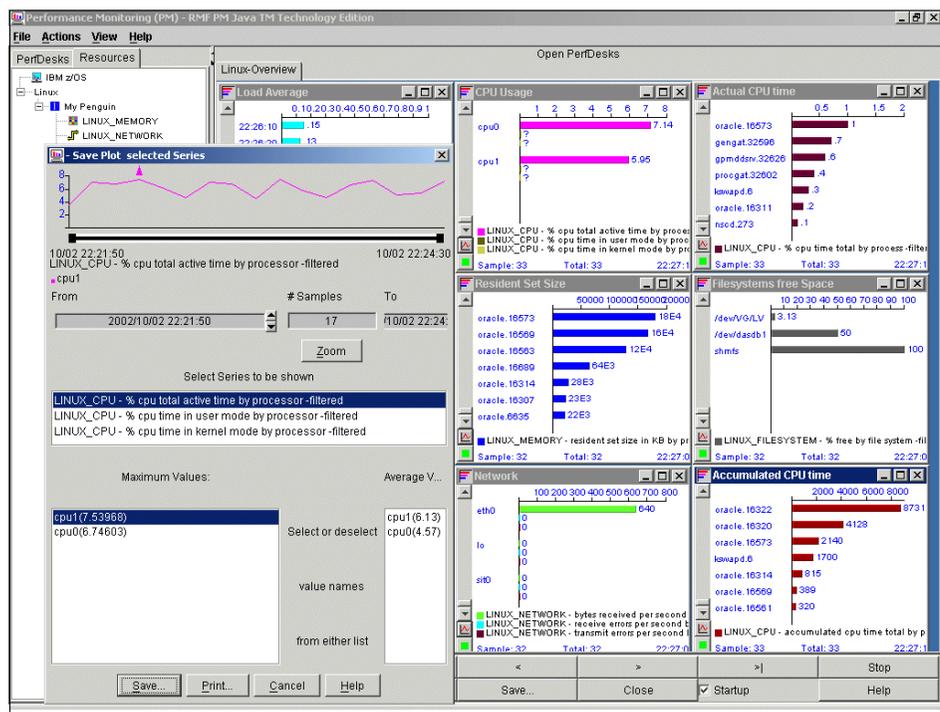


Figure 11-6 RMF performance monitor client

# ESALPS

ESALPS (the Linux Performance Suite) is a suite of products provided by Velocity Software. The four products that make up this suite are:

- ▶ ESAMAP, the VM Monitor Analysis Program, providing performance reports on all aspects of VM/ESA and z/VM performance
- ▶ ESAMON, the VM Real Time Monitor, providing real time analysis of performance
- ▶ ESATCP, the network and Linux data collection program
- ▶ ESAWEB, a very fast VM based Web server.

In addition to the four products, ESALPS provides a Web based interface to view performance data via a Web browser and many control facilities.

## Monitoring requirements

There are many requirements for data collection met by ESALPS. Data is provided for:

- ▶ Capacity Planning: Long term data in the form of a performance database (PDB) is needed as input to long term capacity planning and trend analysis. Full historical data functions are provided with collection, and many forms of data extraction tools.
- ▶ Performance Analysis: Trend data allows the analyst to detect performance changes in any of thousands of potential problem areas. The performance database allows analysts to determine what of many potential changes occurred in the system. Reporting on specific periods of time can be performed allowing in depth performance analysis of performance problems.
- ▶ Real Time Performance: Beyond the traditional entry level real-time performance reporting of the top users and system utilization, real-time performance analysis is provided for all subsystems, user activity, as well as Linux (and many other platforms) servers. Network data is also provided real time.
- ▶ Linux Data: With the advent of virtual Linux server farms on z/VM, performance data is required.

## Standard interface

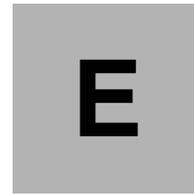
ESALPS uses standard interfaces for all data collection. The advantage to using the standard interfaces provided is that when there are a multitude of releases and distributions available, the standard interfaces will provide consistent data sources.

- ▶ z/VM provides a *monitor interface* that has been available since 1988. Since then, this interface has provided a consistent view of performance of VM systems.
- ▶ Network performance is collected using SNMP, the standard for network management.
- ▶ NETSNMP, an open source software package provides host data for the Linux and other platforms.
- ▶ VM Application data interface is used by applications to insert data into the monitor stream consistent with the standard monitor interface. ESATCP uses this interface to ensure consistent data collection that allows full integration of Linux and VM data.

## Performance Database (PDB)

ESALPS provides both real-time data and historical data for in depth analysis. The performance data is collected daily with a one minute granularity based on the monitor interval. A longer term archive is collected usually with a granularity of 15 minutes. This PDB includes VM data, Linux data and network data.





## **Useful Linux information**

This appendix is intended for those who are not familiar with Linux, but are beginning to use it for the first time to run Linux guests on zSeries.

## Useful vi command

The vi command works in two modes: input and command mode. To switch between the two you can use the escape key, then do the following:

- ▶ **-i** puts you in input mode
  - esc** - brings you to command mode
- ▶ **oO** command mode - Esc puts you in command mode
  - o i** - brings you back to input mode
  - x** Delete a character
  - D** Delete to the end of the line
  - dd** deletes a line and puts it in the buffer (3dd will delete 3 lines)
  - u** undo last delete
  - p** add the buffer past the current location
  - P** add the buffer before the current location
  - o** add a line and go into insert mode
  - O** add a line above the current line
  - /string** search for string
  - n** search again
  - :** go to command prompt
  - hjk<sup>l</sup>** cursor movement (but arrow keys usually work)  
h- left, j-down, k-up, l-right
  - yy** yank a link - put it in the buffer (3yy yanks 3 lines)
  - A** add text at the end of the line
  - :1,\$** change first variable to second variable, for example,  
:**%s/this/that/g**  
**1,\$s/this/that/g**  
If you put yourself at the end of the file it will change every "this" to "that"
  - :wq** write and quit
  - :q!** quit and discard changes
  - .** repeat the last command

# Useful Linux commands

These Linux commands could be useful:

<b>cat</b>	Concatenate files and print on the standard output
<b>cd</b>	Change working directory <b>cd -</b> - takes you back to the previous directory
<b>chmod</b>	Change file access permissions
<b>chown</b>	Change ownership of files
<b>clear</b>	Clear the terminal screen
<b>cp</b>	Copy files or directories
<b>dasdfmt</b>	Formats a DASD (ECKD) disk drive to prepare it for usage with Linux for zSeries
<b>dmesg</b>	Print or control the kernel ring buffer
<b>fdasd</b>	Partitioning tool for DASDs
<b>fuser</b>	Identify processes using files or sockets
<b>grep</b>	Print lines matching a pattern
<b>ifconfig</b>	Configure a network interface
<b>insmod</b>	Load a module into the Linux kernel.
<b>kill</b>	Terminate a process
<b>killall</b>	Terminate processes by name
<b>less</b>	Filter for paging through text one screen at a time (more powerful than more)
<b>ls</b>	List a directory contents
<b>man</b>	Help command
<b>mkdir</b>	Make a new directory
<b>mke2fs</b>	Create an ext2 file system.
<b>more</b>	Filter for paging through text one screen at a time
<b>mount</b>	Mount a file system mount -o loo
<b>mv</b>	Move (rename) file
<b>pidof</b>	Find the process ID of a running program
<b>pwd</b>	Print name of current/working directory
<b>scp</b>	Copy files between two machines. It uses ssh for the file transfer.
<b>silo</b>	Preparing a DASD to become an IPL device (also ziph)
<b>ssh</b>	Tool to log into another machine or execute commands remotely.
<b>su</b>	Change user and group ID
<b>tail</b>	Print the last lines of a file. Often used with the -f parameter to watch log files
<b>traceroute</b>	Utility to print the route of packets through the network
<b>yast</b>	Configuration utility for SuSE Linux

# Linux directory structure

This is a description of the 14 directories used in Linux:

/bin	contains executables for every user
/boot	contains the kernel image, system map, bootstrap, etc
/dev	contains special files that represent hardware devices
/etc	contains configuration files
profile	default settings for the system
fstab	file system table, default mounts
X11	X-Window configuration files
rc.d	startup scripts
inetd.conf	inetd - listener configuration
skel	directory containing default user configuration files
passwd	user information and passwords
shadow	user passwords (when in use, no password in the passwd file)
/home	contains users' home directories
/lib	contains libraries
/mnt	contains mount points for other file systems
/proc	virtual file system (in memory) contains kernel and process information
cpuinfo	CPU information
dasd/devices	dasd addresses that the system recognized
devices	devices supported
meminfo	memory utilization
proc/swaps	swap file or devices enabled
proc/uptime	time since last boot
proc/version	Kernel version and gcc level
/root	home directory for root user
/sbin	contains system admin executables (superuser bin)
/tmp	temporary storage space
/usr	UNIX System Resources - contains all non-essential programs and libraries which can be either NFS mount or read only
/var	contains log and spool files, temp storage space (other than /tmp)
/lost+found	location for lost+found files are stored after a crash recovery by fs2ck

## Important Linux configuration files

/etc/rc.config

/etc/ssh/sshd.config

/etc/inittab

/etc/fstab

/etc/zilo.conf

/etc/(x)inetd.conf

## Tools to administrate Linux

To administrate your Linux over the network, you should no longer use insecure protocols like Telnet. Therefore, the Telnet server that is installed during a standard SLES 7 installation should be disabled afterwards. Instead, use the also installed ssh server and connect to it via an ssh client. On Linux machines you mainly have already a ssh client installed. If you want to administrate your Linux machines from another platform than Linux, you have to find an appropriate client for this platform. One example for the windows platform is putty.

To transfer files to your Linux machine in the past FTP was mainly used. Just like in a Telnet session, in a FTP session, the password and the data is transferred without any encryption. To secure your server, you should not use FTP for the file transfer. Instead, you can also make use of ssh and use secure copy (scp).

### Putty

Putty is a free Telnet and ssh client for windows platforms. You can download the program and find documentation on the following Web site:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>

### WinSCP

WinSCP is free secure copy client for windows platforms. It has a graphical interface to simplify transferring files to another host. To download it following the links of the products homepage:

<http://www.winscp.vse.cz/eng/index.php.com>





## Sample of response file

This appendix shows the modified response file that we used in 3.15, “Using the non-interactive install process” on page 61.

The lines we changed are:

- ▶ `UNIX_GROUP_NAME=dba`
- ▶ `FROM_LOCATION=/oracle/Disk1/stage/products.jar`
- ▶ `ORACLE_HOME=/oracle/OraHome1`
- ▶ `ORACLE_HOME_NAME=OraHome1`
- ▶ `#DEINSTALL_LIST={'oracle.server','9.2.0.1.0'}`
- ▶ `SHOW_SPLASH_SCREEN=false`
- ▶ `SHOW_WELCOME_PAGE=false`
- ▶ `SHOW_COMPONENT_LOCATIONS_PAGE=false`
- ▶ `SHOW_CUSTOM_TREE_PAGE=false`
- ▶ `SHOW_SUMMARY_PAGE=false`
- ▶ `SHOW_INSTALL_PROGRESS_PAGE=false`
- ▶ `SHOW_REQUIRED_CONFIG_TOOL_PAGE=false`
- ▶ `SHOW_OPTIONAL_CONFIG_TOOL_PAGE=false`
- ▶ `SHOW_RELEASE_NOTES=false`
- ▶ `SHOW_ROOTSH_CONFIRMATION=false`
- ▶ `SHOW_END_SESSION_PAGE=false`
- ▶ `SHOW_EXIT_CONFIRMATION=false`
- ▶ `NEXT_SESSION=false`
- ▶ `#Parameter: LOCATION_DISK2`
- ▶ `#Example : NEXT_SESSION = true`

- ▶ LOCATION\_FOR\_DISK2=/oracle/Disk2
- ▶ NEXT\_SESSION\_ON\_FAIL=false
- ▶ SHOW\_DEINSTALL\_CONFIRMATION=false
- ▶ SHOW\_DEINSTALL\_PROGRESS=false
- ▶ s\_GlobalDBName=ora92.us.oracle.com
- ▶ s\_oracleSID=ora92
- ▶ JDK\_HOME=/oracle/IBMJava2-s390-131/
- ▶ s\_mountPoint=/oracle/oradata
- ▶ JDK\_HOME=/oracle/IBMJava2-s390-131

This is the response file which is /oracle/response/ora92.rsp:

```
#####
## Copyright(c) Oracle Corporation 1998,2002. All rights reserved.##
##
## Specify values for the variables listed below to customize      ##
## your installation.                                           ##
##
## Each variable is associated with a comment. The comment      ##
## identifies the variable type.                                ##
##
## Please specify the values in the following format:           ##
##
##      Type      Example
##      String    "Sample Value"
##      Boolean   True or False
##      Number    1000
##      StringList {"String value 1","String Value 2"}
##
## The values that are given as <Value Required> need to be    ##
## specified for a silent installation to be successful.      ##
##
##
## This response file is generated by Oracle Software          ##
## Packager.                                                  ##
#####
```

```
[General]
RESPONSEFILE_VERSION=1.7.0
```

```
[SESSION]
#Parameter: UNIX_GROUP_NAME
#Type: String
#Description: Unix group to be set for the inventory directory. Valid only
in Unix platforms.
#Example : UNIX_GROUP_NAME = "install"
```

*UNIX\_GROUP\_NAME=dba*

#Parameter: FROM\_LOCATION  
#Type: String  
#Description: Complete path of the products.jar.  
#Example : FROM\_LOCATION = "../stage/products.jar"  
*FROM\_LOCATION=/oracle/Disk1/stage/products.jar*

#Parameter: FROM\_LOCATION\_CD\_LABEL  
#Type: String  
#Description: This variable should only be used in multi-CD installations. It includes the label of the Compact Disk where the file "products.jar" exists. The label can be found in the file "disk.label" in the same directory as products.jar.  
#Example : FROM\_LOCATION\_CD\_LABEL = "CD Label"  
FROM\_LOCATION\_CD\_LABEL=<Value Unspecified>

#Parameter: NEXT\_SESSION\_RESPONSE  
#Type: String  
#Description: Optionally specifies the full path of next session's response file. If only a file name is specified, the response file is retrieved from <TEMP>/oraInstall directory. This variable is only active if NEXT\_SESSION is set to true.  
#Example : NEXT\_SESSION\_RESPONSE = "nextinstall.rsp"  
NEXT\_SESSION\_RESPONSE=<Value Unspecified>

#Parameter: ORACLE\_HOME  
#Type: String  
#Description: Complete Location of the Oracle Home.  
#Example : ORACLE\_HOME = "C:\OHOME1"  
*ORACLE\_HOME=/oracle/OraHome1*

#Parameter: ORACLE\_HOME\_NAME  
#Type: String  
#Description: Oracle Home Name. Used in creating folders, services.  
#Example : ORACLE\_HOME\_NAME = "OHOME1"  
*ORACLE\_HOME\_NAME=OraHome1*

#Parameter: TOPLEVEL\_COMPONENT  
#Type: StringList  
#Description: The Toplevel component that has to be installed in the current session.  
#The following choices are available. The value should contain only one of these choices.  
#The choices are of the form Internal Name, Version : External name. Please use the internal name and version while specifying the value.  
# oracle.server, 9.2.0.1.0 : Oracle9i Database 9.2.0.1.0  
# oracle.client, 9.2.0.1.0 : Oracle9i Client 9.2.0.1.0

```

# oracle.infrastructure, 9.2.0.1.0 : Oracle9i Management and Integration
9.2.0.1.0
# oracle.options.ops.clustermgr, 9.2.0.1.0 : Oracle Cluster Manager
9.2.0.1.0
#Example : TOPLEVEL_COMPONENT = {"oracle.server","9.2.0.1.0"}
TOPLEVEL_COMPONENT={"oracle.server","9.2.0.1.0"}

#Parameter: DEINSTALL_LIST
#Type: StringList
#Description: List of components to be deinstalled during a deinstall
session.
#The following choices are available. The value should contain only one of
these choices.
#The choices are of the form Internal Name, Version : External name. Please
use the internal name and version while specifying the value.
# oracle.server, 9.2.0.1.0 : Oracle9i Database 9.2.0.1.0
# oracle.client, 9.2.0.1.0 : Oracle9i Client 9.2.0.1.0
# oracle.infrastructure, 9.2.0.1.0 : Oracle9i Management and Integration
9.2.0.1.0
# oracle.options.ops.clustermgr, 9.2.0.1.0 : Oracle Cluster Manager
9.2.0.1.0
#Example : DEINSTALL_LIST = {"oracle.server","9.2.0.1.0"}
#DEINSTALL_LIST={"oracle.server","9.2.0.1.0"}

#Parameter: SHOW_SPLASH_SCREEN
#Type: Boolean
#Description: Set to true if the initial splash screen in the installer
needs to be shown.
#Example : SHOW_SPLASH_SCREEN = true
SHOW_SPLASH_SCREEN=false

#Parameter: SHOW_WELCOME_PAGE
#Type: Boolean
#Description: Set to true if the Welcome page in the installer needs to be
shown.
#Example : SHOW_WELCOME_PAGE = false
SHOW_WELCOME_PAGE=false

#Parameter: SHOW_COMPONENT_LOCATIONS_PAGE
#Type: Boolean
#Description: Set to true if the component locations page in the installer
needs to be shown.
#This page only appears if there are products whose installed directory can
be changed.
#If you set this to false you will prevent the user from being able to
specify alternate directories.
#Example : SHOW_COMPONENT_LOCATIONS_PAGE = false
SHOW_COMPONENT_LOCATIONS_PAGE=false

```

#Parameter: SHOW\_CUSTOM\_TREE\_PAGE  
#Type: Boolean  
#Description: Set to true if the custom tree page in the installer needs to be shown.  
#In this page dependencies can be selected or unselected. This page appears only in a custom install type.  
#Example : SHOW\_CUSTOM\_TREE\_PAGE = false  
SHOW\_CUSTOM\_TREE\_PAGE=false

#Parameter: SHOW\_SUMMARY\_PAGE  
#Type: Boolean  
#Description: Set to true if the summary page in the installer needs to be shown.  
#The summary page shows the list of components that will be installed in this session.  
#Example : SHOW\_SUMMARY\_PAGE = true  
SHOW\_SUMMARY\_PAGE=false

#Parameter: SHOW\_INSTALL\_PROGRESS\_PAGE  
#Type: Boolean  
#Description: Set to true if the install progress page in the installer needs to be shown.  
#This page shows the current status in the installation. The current status includes which product is being installed, which file is being copied.  
#Example : SHOW\_INSTALL\_PROGRESS\_PAGE = true  
SHOW\_INSTALL\_PROGRESS\_PAGE=false

#Parameter: SHOW\_REQUIRED\_CONFIG\_TOOL\_PAGE  
#Type: Boolean  
#Description: Set to true if the required config tools page in the installer needs to be shown.  
#This page shows the list of required configuration tools that are part of this installation.  
#It shows the status of each tool, including any failures with detailed information on why the tool has failed.  
#Example : SHOW\_REQUIRED\_CONFIG\_TOOL\_PAGE = true  
SHOW\_REQUIRED\_CONFIG\_TOOL\_PAGE=false

#Parameter: SHOW\_OPTIONAL\_CONFIG\_TOOL\_PAGE  
#Type: Boolean  
#Description: Set to true if the optional config tools page in the installer needs to be shown.  
#This page shows the list of optional configuration tools that are part of this installation and are configured to launch automatically.  
#It shows the status of each tool, including any failures with detailed information on why the tool has failed.  
#Example : SHOW\_OPTIONAL\_CONFIG\_TOOL\_PAGE = true  
SHOW\_OPTIONAL\_CONFIG\_TOOL\_PAGE=false

```
#Parameter: SHOW_RELEASE_NOTES
#Type: Boolean
#Description: Set to true if the release notes of this installation need to
be shown at the end of installation.
#This dialog is launchable from the End of Installation page and shows the
list of release notes available for the products just installed.
# This also requires the variable SHOW_END_SESSION_PAGE variable to be set
to true.
#Example : SHOW_RELEASE_NOTES = true
SHOW_RELEASE_NOTES=false

#Parameter: SHOW_ROOTSH_CONFIRMATION
#Type: Boolean
#Description: Set to true if the Confirmation dialog asking to run the
root.sh script in the installer needs to be shown.
#Valid only in Unix platforms.
#Example : SHOW_ROOTSH_CONFIRMATION = true
SHOW_ROOTSH_CONFIRMATION=false

#Parameter: SHOW_END_SESSION_PAGE
#Type: Boolean
#Description: Set to true if the end of session page in the installer needs
to be shown.
#This page shows if the installation is successful or not.
#Example : SHOW_END_SESSION_PAGE = true
SHOW_END_SESSION_PAGE=false

#Parameter: SHOW_EXIT_CONFIRMATION
#Type: Boolean
#Description: Set to true if the confirmation when exiting the installer
needs to be shown.
#Example : SHOW_EXIT_CONFIRMATION = true
SHOW_EXIT_CONFIRMATION=false

#Parameter: NEXT_SESSION
#Type: Boolean
#Description: Set to true to allow users to go back to the File Locations
page for another installation. This flag also needs to be set to true in
order to process another response file (see NEXT_SESSION_RESPONSE).
#Example : NEXT_SESSION = true
NEXT_SESSION=false

#Parameter: LOCATION_DISK2
#Example : NEXT_SESSION = true
LOCATION_FOR_DISK2=/oracle/Disk2

#Parameter: NEXT_SESSION_ON_FAIL
#Type: Boolean
```

#Description: Set to true to allow users to invoke another session even if current install session has failed. This flag is only relevant if NEXT\_SESSION is set to true.

#Example : NEXT\_SESSION\_ON\_FAIL = true  
NEXT\_SESSION\_ON\_FAIL=false

#Parameter: SHOW\_DEINSTALL\_CONFIRMATION

#Type: Boolean

#Description: Set to true if deinstall confirmation is needed during a deinstall session.

#Example : SHOW\_DEINSTALL\_CONFIRMATION = true  
SHOW\_DEINSTALL\_CONFIRMATION=false

#Parameter: SHOW\_DEINSTALL\_PROGRESS

#Type: Boolean

#Description: Set to true if deinstall progress is needed during a deinstall session.

#Example : SHOW\_DEINSTALL\_PROGRESS = true  
SHOW\_DEINSTALL\_PROGRESS=false

[oracle.server\_9.2.0.1.0]

#Parameter: COMPONENT\_LANGUAGES

#Type: StringList

#Description: Languages in which the components will be installed.

#The following choices are available. The value should contain only one of these choices.

#The choices are of the form Internal Name : External name. Please use the internal name while specifying the value.

# en, : English  
# fr, : French  
# ar, : Arabic  
# bn, : Bengali  
# pt\_BR, : Brazilian Portuguese  
# bg, : Bulgarian  
# fr\_CA, : Canadian French  
# ca, : Catalan  
# hr, : Croatian  
# cs, : Czech  
# da, : Danish  
# nl, : Dutch  
# ar\_EG, : Egyptian  
# en\_GB, : English (United Kingdom)  
# et, : Estonian  
# fi, : Finnish  
# de, : German  
# el, : Greek  
# iw, : Hebrew  
# hu, : Hungarian

```

# is,    : Icelandic
# in,    : Indonesian
# it,    : Italian
# ja,    : Japanese
# ko,    : Korean
# es,    : Latin American Spanish
# lv,    : Latvian
# lt,    : Lithuanian
# ms,    : Malay
# es_MX, : Mexican Spanish
# no,    : Norwegian
# pl,    : Polish
# pt,    : Portuguese
# ro,    : Romanian
# ru,    : Russian
# zh_CN, : Simplified Chinese
# sk,    : Slovak
# sl,    : Slovenian
# es_ES, : Spanish
# sv,    : Swedish
# th,    : Thai
# zh_TW, : Traditional Chinese
# tr,    : Turkish
# uk,    : Ukrainian
# vi,    : Vietnamese
#Example : COMPONENT_LANGUAGES = {"en"}
COMPONENT_LANGUAGES={"en"}

```

```

#Parameter: INSTALL_TYPE
#Type: String
#Description: Installation type of the component.
#The following choices are available. The value should contain only one of
these choices.
#The choices are of the form Internal Name : External name. Please use the
internal name while specifying the value.
# EE,    : Enterprise Edition
# SE,    : Standard Edition
# Custom, : Custom
#Example : INSTALL_TYPE = "EE"
INSTALL_TYPE="EE"

```

```

#Parameter: s_serverInstallType
#Type: String
s_serverInstallType=<Value Unspecified>

```

```

#Parameter: s_selectedNodes
#Type: String
s_selectedNodes=<Value Unspecified>

```

#Parameter: s\_dbcaProgressOnly  
#Type: String  
s\_dbcaProgressOnly=<Value Unspecified>

#Parameter: s\_cfgtyperet  
#Type: String  
s\_cfgtyperet=<Value Unspecified>

#Parameter: s\_bundleName  
#Type: String  
s\_bundleName=<Value Unspecified>

#Parameter: b\_rdbmsInstalling  
#Type: Boolean  
b\_rdbmsInstalling=<Value Unspecified>

#Parameter: b\_launchNETCA  
#Type: Boolean  
b\_launchNETCA=<Value Unspecified>

#Parameter: b\_autoStartApache  
#Type: Boolean  
b\_autoStartApache=<Value Unspecified>

[oracle.options\_9.2.0.1.0]  
#Parameter: s\_serverInstallType  
#Type: String  
s\_serverInstallType=<Value Unspecified>

#Parameter: s\_cfgtyperet  
#Type: String  
s\_cfgtyperet=<Value Unspecified>

#Parameter: s\_bundleName  
#Type: String  
s\_bundleName=<Value Unspecified>

[oracle.options.ops\_9.2.0.1.0]  
#Parameter: s\_serverInstallType  
#Type: String  
s\_serverInstallType=<Value Unspecified>

#Parameter: s\_cfgtyperet  
#Type: String  
s\_cfgtyperet=<Value Unspecified>

#Parameter: s\_OPSSelectedNodes

```
#Type: String  
s_OPSSelectedNodes=<Value Unspecified>
```

```
#Parameter: s_rawDeviceName  
#Type: String  
s_rawDeviceName=<Value Unspecified>
```

```
[oracle.cartridges.spatial_9.2.0.1.0]  
#Parameter: s_bundleName  
#Type: String  
s_bundleName=<Value Unspecified>
```

```
[oracle.options.ano_9.2.0.1.0]  
#Parameter: s_bundleName  
#Type: String  
s_bundleName=<Value Unspecified>
```

```
#Parameter: s_OPSSelectedNodes  
#Type: String  
s_OPSSelectedNodes=<Value Unspecified>
```

```
[oracle.options.odm_9.2.0.1.0]  
#Parameter: s_OPSSelectedNodes  
#Type: String  
s_OPSSelectedNodes=<Value Unspecified>
```

```
[oracle.rdbms_9.2.0.1.0]  
#Parameter: s1_dbaOperGroups  
#Type: StringList  
s1_dbaOperGroups=<Value Unspecified>
```

```
#Parameter: s_serverInstallType  
#Type: String  
s_serverInstallType=<Value Unspecified>
```

```
#Parameter: s_nameOfBundle  
#Type: String  
s_nameOfBundle=<Value Unspecified>
```

```
#Parameter: s_dbcaProgressOnly  
#Type: String  
s_dbcaProgressOnly=<Value Unspecified>
```

```
#Parameter: s_cfgtyperet  
#Type: String
```



```
#Type: Boolean
b_launchNETCA=<Value Unspecified>
```

```
[oracle.emprod_9.2.0.1.0]
#Parameter: s_cfgtyperet
#Type: String
s_cfgtyperet=<Value Unspecified>
```

```
#Parameter: b_launchEMCA
#Type: Boolean
b_launchEMCA=<Value Unspecified>
```

```
[oracle.assistants.dbma_9.2.0.1.0]
#Parameter: OPTIONAL_CONFIG_TOOLS
#Type: StringList
#Description: List of Optional Config tools that needs to be launched.
#The following choices are available. The value should contain only one of
these choices.
#The choices are of the form Internal Name : External name. Please use the
internal name while specifying the value.
# dbma, : Database Upgrade Assistant
#Example : OPTIONAL_CONFIG_TOOLS = {"dbma"}
OPTIONAL_CONFIG_TOOLS=<Value Unspecified>
```

```
#Parameter: sl_migrateSIDDIALOG_RETURN
#Type: StringList
sl_migrateSIDDIALOG_RETURN=<Value Unspecified>
```

```
#Parameter: s_sidToMigrate
#Type: String
s_sidToMigrate=<Value Unspecified>
```

```
#Parameter: s_cfgtyperet
#Type: String
s_cfgtyperet=<Value Unspecified>
```

```
#Parameter: b_rdbmsInstalling
#Type: Boolean
b_rdbmsInstalling=<Value Unspecified>
```

```
#Parameter: b_noMigration
#Type: Boolean
b_noMigration=<Value Unspecified>
```

```
[oracle.utilities.util_9.2.0.1.0]
#Parameter: s_OPSSelectedNodes
```

#Type: String  
s\_OPSSelectedNodes=<Value Unspecified>

#Parameter: b\_rdbmsInstalling  
#Type: Boolean  
b\_rdbmsInstalling=<Value Unspecified>

[oracle.options.intermedia.imserver\_9.2.0.1.0]  
#Parameter: s\_bundleName  
#Type: String  
s\_bundleName=<Value Unspecified>

[oracle.sysman.oms\_9.2.0.1.0]  
#Parameter: OPTIONAL\_CONFIG\_TOOLS  
#Type: StringList  
#Description: List of Optional Config tools that needs to be launched.  
#The following choices are available. The value should contain only one of these choices.  
#The choices are of the form Internal Name : External name. Please use the internal name while specifying the value.  
# emca, : Enterprise Manager Configuration Assistant  
#Example : OPTIONAL\_CONFIG\_TOOLS = {"emca"}  
OPTIONAL\_CONFIG\_TOOLS=<Value Unspecified>

#Parameter: s\_emca\_responseFileName  
#Type: String  
s\_emca\_responseFileName=<Value Unspecified>

#Parameter: launchEMCA  
#Type: Boolean  
launchEMCA=<Value Unspecified>

#Parameter: isTypicalInstall  
#Type: Boolean  
isTypicalInstall=<Value Unspecified>

#Parameter: isSilentMode  
#Type: Boolean  
isSilentMode=<Value Required>

#Parameter: ServerRepository\_index  
#Type: Number  
ServerRepository\_index=<Value Required>

#Parameter: AUTOSVC  
#Type: String  
AUTOSVC=<Value Unspecified>

```
[oracle.cartridges.locator_9.2.0.1.0]
#Parameter: s_bundleName
#Type: String
s_bundleName=<Value Unspecified>
```

```
[oracle.isearch.server_9.2.0.1.0]
#Parameter: b_iAS
#Type: Boolean
b_iAS=<Value Unspecified>
```

```
[oracle.options.ano.sns_9.2.0.1.0]
#Parameter: s_bundleName
#Type: String
s_bundleName=<Value Unspecified>
```

```
[oracle.java.javavm_9.2.0.1.0]
#Parameter: b_javavmIsInstalling
#Type: Boolean
b_javavmIsInstalling=<Value Unspecified>
```

```
[oracle.networking.netsrv_9.2.0.1.0]
#Parameter: b_net8ServerIsInstalling
#Type: Boolean
b_net8ServerIsInstalling=<Value Unspecified>
```

```
[oracle.cartridges.context_9.2.0.1.0]
#Parameter: s_OPSSelectedNodes
#Type: String
s_OPSSelectedNodes=<Value Unspecified>
```

```
[oracle.apache_9.2.0.1.0]
#Parameter: OPTIONAL_CONFIG_TOOLS
#Type: StringList
#Description: List of Optional Config tools that needs to be launched.
#The following choices are available. The value should contain only one of
these choices.
#The choices are of the form Internal Name : External name. Please use the
internal name while specifying the value.
# configtool1, : Starting HTTP Server
# configtool2, : Starting Oracle HTTP service
#Example : OPTIONAL_CONFIG_TOOLS = {"configtool1"}
```

```
OPTIONAL_CONFIG_TOOLS=<Value Unspecified>

#Parameter: s_oracleSID
#Type: String
#Description: Oracle SID that can be passed in from top level install
component.
s_oracleSID=ora92

#Parameter: s_jservPort
#Type: String
#Description: The port Apache JServ listens to
s_jservPort=<Value Unspecified>

#Parameter: s_apacheVersionNumber
#Type: String
#Description: version number of apache passed to oem agent
s_apacheVersionNumber=<Value Unspecified>

#Parameter: s_apachePortSSL
#Type: String
#Description: Apache SSL port used when starting in ssl mode. Default 443
s_apachePortSSL=<Value Unspecified>

#Parameter: s_apachePortNonSSL
#Type: String
#Description: Port number that apache uses when starting in non-ssl mode
s_apachePortNonSSL=<Value Unspecified>

#Parameter: s_apachePort
#Type: String
#Description: Port number that apache uses when starting in ssl mode
s_apachePort=<Value Unspecified>

#Parameter: s_NLSLANG
#Type: String
#Description: String to hold the NLS LANG value determined by
GetNlsLangWindows q
#uery.
s_NLSLANG=<Value Required>

#Parameter: s_LANGUAGE_TERRITORY
#Type: String
#Description: Substring of s_NLSLANG
s_LANGUAGE_TERRITORY=<Value Required>

#Parameter: b_autoStartApache
#Type: Boolean
#Description: Set to false if you do not want apache to start after
installation
```

```

b_autoStartApache=<Value Unspecified>

#Parameter: b_apacheInstalling
#Type: Boolean
#Description: Set to true and passed to oem agent
b_apacheInstalling=<Value Unspecified>

#Parameter: s_topDir
#Type: String
#Description: APACHE_TOP directory for Apache Web Server
s_topDir=<Value Unspecified>

#Parameter: s_jvm
#Type: String
#Description: Path to the Java Virtual Machine
s_jvm=<Value Unspecified>

#Parameter: ServerRoot
#Type: String
#Description: Apache Server root directory
ServerRoot=<Value Unspecified>

#Parameter: JDK_HOME
#Type: String
#Description: JDK home location
JDK_HOME=/oracle/IBMJava2-s390-131/

#Parameter: APACHE_HOME
#Type: String
#Description: Main apache directory off of ORACLE_HOME where all components
live under i.e. ORACLE_HOME/Apache
APACHE_HOME=<Value Unspecified>

#Parameter: s_oracleApacheConfigFile
#Type: String
#Description: Location of oracle_apache.conf file to be passed to all
modules that need to include there conf file.
s_oracleApacheConfigFile=<Value Unspecified>

#Parameter: s_oracleJservPropertiesFile
#Type: String
#Description: Location of the Jserv.properties file
s_oracleJservPropertiesFile=<Value Unspecified>

[oracle.soap.jserv_2.0.0.0.0a]
#Parameter: s_soapPort
#Type: String
#Description: The port that SOAP/JServ listens to

```

s\_soapPort=<Value Unspecified>

[oracle.rdbms.nid\_9.2.0.1.0]  
#Parameter: s\_OPSSelectedNodes  
#Type: String  
s\_OPSSelectedNodes=<Value Unspecified>

[oracle.emprod.agent\_ext.emd\_agentext\_9.2.0.1.0]  
#Parameter: s\_OPSSelectedNodes  
#Type: String  
s\_OPSSelectedNodes=<Value Unspecified>

[oracle.rdbms.ds\_9.2.0.1.0]  
#Parameter: s\_OPSSelectedNodes  
#Type: String  
s\_OPSSelectedNodes=<Value Unspecified>

[oracle.isearch.is\_common\_9.2.0.1.0]  
#Parameter: b\_iAS  
#Type: Boolean  
b\_iAS=<Value Unspecified>

[oracle.emprod.agent\_ext.ows\_agentext\_9.2.0.1.0]  
#Parameter: s\_apacheVersionNumber  
#Type: String  
s\_apacheVersionNumber=<Value Unspecified>

#Parameter: b\_apacheInstalling  
#Type: Boolean  
b\_apacheInstalling=<Value Unspecified>

[oracle.emprod.oemagent\_9.2.0.1.0]  
#Parameter: s\_cfgtyperet  
#Type: String  
s\_cfgtyperet=<Value Unspecified>

[oracle.emprod.oemagent.agentca\_9.2.0.1.0]  
#Parameter: OPTIONAL\_CONFIG\_TOOLS  
#Type: StringList  
#Description: List of Optional Config tools that needs to be launched.  
#The following choices are available. The value should contain only one of these choices.

```

#The choices are of the form Internal Name : External name. Please use the
internal name while specifying the value.
# agentca, : Agent Configuration Assistant
#Example : OPTIONAL_CONFIG_TOOLS = {"agentca"}
OPTIONAL_CONFIG_TOOLS=<Value Unspecified>

#Parameter: s_cfgtyperet
#Type: String
s_cfgtyperet=<Value Unspecified>

#Parameter: b_launchAgentCA
#Type: Boolean
b_launchAgentCA=<Value Unspecified>

[oracle.assistants.dbca_9.2.0.1.0]
#Parameter: OPTIONAL_CONFIG_TOOLS
#Type: StringList
#Description: List of Optional Config tools that needs to be launched.
#The following choices are available. The value should contain only one of
these choices.
#The choices are of the form Internal Name : External name. Please use the
internal name while specifying the value.
# dbca, : Oracle Database Configuration Assistant
#Example : OPTIONAL_CONFIG_TOOLS = {"dbca"}
OPTIONAL_CONFIG_TOOLS=<Value Unspecified>

#Parameter: s_serverInstallType
#Type: String
s_serverInstallType=<Value Unspecified>

#Parameter: s_responseFileName
#Type: String
s_responseFileName=<Value Unspecified>

#Parameter: s_oidPasswd
#Type: String
s_oidPasswd=<Value Unspecified>

#Parameter: s_oidAdmin
#Type: String
s_oidAdmin=<Value Unspecified>

#Parameter: s_instType
#Type: String
s_instType=<Value Unspecified>

#Parameter: s_globalDBName
#Type: String

```

```
s_globalDBName=<Value Unspecified>

#Parameter: s_dbRetChoice
#Type: String
s_dbRetChoice=<Value Unspecified>

#Parameter: s_dbRetChar
#Type: String
s_dbRetChar=<Value Unspecified>

#Parameter: s_cfgtyperet
#Type: String
s_cfgtyperet=<Value Unspecified>

#Parameter: ps_dbCharSet
#Type: String
ps_dbCharSet=<Value Unspecified>

#Parameter: pb_askMountPoint
#Type: Boolean
pb_askMountPoint=<Value Unspecified>

#Parameter: b_showCharsetDialog
#Type: Boolean
b_showCharsetDialog=<Value Unspecified>

#Parameter: b_rdbmsInstalling
#Type: Boolean
b_rdbmsInstalling=<Value Unspecified>

#Parameter: b_noMigration
#Type: Boolean
b_noMigration=<Value Unspecified>

#Parameter: b_lowResource
#Type: Boolean
b_lowResource=<Value Unspecified>

#Parameter: b_iAS
#Type: Boolean
b_iAS=<Value Unspecified>

#Parameter: b_createStarterDBReturn
#Type: Boolean
b_createStarterDBReturn=<Value Unspecified>

#Parameter: b_configureOid
#Type: Boolean
b_configureOid=<Value Unspecified>
```

```
#Parameter: CLUSTER_SERVICES
#Type: String
CLUSTER_SERVICES=<Value Unspecified>

#Parameter: s_dbcaProgressOnly
#Type: String
s_dbcaProgressOnly=<Value Unspecified>

#Parameter: s_cfgname
#Type: String
s_cfgname=<Value Unspecified>

#Parameter: pn_softwareSize
#Type: Number
pn_softwareSize=<Value Unspecified>

#Parameter: b_passwdDialog
#Type: Boolean
b_passwdDialog=<Value Unspecified>

#Parameter: s_seedLocation
#Type: String
s_seedLocation=<Value Unspecified>

#Parameter: ps_mountPoint
#Type: String
ps_mountPoint=<Value Unspecified>

#Parameter: pn_databaseSize
#Type: Number
pn_databaseSize=<Value Unspecified>

#Parameter: s_templateValue
#Type: String
s_templateValue=<Value Unspecified>

#Parameter: s_dbSid
#Type: String
s_dbSid=ora92

#Parameter: s_mountPoint
#Type: String
s_mountPoint=/oracle/oradata

[oracle.soap.srv_2.0.0.0.0a]
#Parameter: b_serverInstalled
#Type: Boolean
```

#Description: set to true and passed to client when server is installed.  
b\_serverInstalled=<Value Unspecified>

[oracle.soap.cli\_2.0.0.0a]  
#Parameter: b\_serverInstalled  
#Type: Boolean  
#Description: Bool set to true if soap server is installed.  
b\_serverInstalled=<Value Unspecified>

#Parameter: s\_hostPort  
#Type: String  
#Description: Host and Port number used in url to soap server  
s\_hostPort=<Value Unspecified>

[oracle.bc4j\_9.0.2.692.1]  
#Parameter: s\_oracleJservPropertiesFile  
#Type: String  
#Description: Location of the JServ Properties file  
s\_oracleJservPropertiesFile=<Value Unspecified>

#Parameter: s\_oc4jdeployini  
#Type: String  
#Description: Refer to deploy.ini under OH\j2ee\deploy.ini  
s\_oc4jdeployini=<Value Unspecified>

#Parameter: APACHE\_HOME  
#Type: String  
#Description: Apache Home directory.  
APACHE\_HOME=<Value Unspecified>

[oracle.rdbms.ovm\_9.2.0.1.0]  
#Parameter: s\_OPSSelectedNodes  
#Type: String  
s\_OPSSelectedNodes=<Value Unspecified>

[oracle.sysman.emcommon\_9.2.0.1.0]  
#Parameter: OPTIONAL\_CONFIG\_TOOLS  
#Type: StringList  
#Description: List of Optional Config tools that needs to be launched.  
#The following choices are available. The value should contain only one of these choices.  
#The choices are of the form Internal Name : External name. Please use the internal name while specifying the value.  
# console, : LaunchOEM  
#Example : OPTIONAL\_CONFIG\_TOOLS = {"console"}

OPTIONAL\_CONFIG\_TOOLS=<Value Unspecified>

[oracle.networking.netclt\_9.2.0.1.0]

#Parameter: s\_cfgtyperet

#Type: String

s\_cfgtyperet=<Value Unspecified>

#Parameter: s\_bundleName

#Type: String

s\_bundleName=<Value Unspecified>

#Parameter: b\_rdbmsInstalling

#Type: Boolean

b\_rdbmsInstalling=<Value Unspecified>

#Parameter: b\_net8ServerInstalling

#Type: Boolean

b\_net8ServerInstalling=<Value Unspecified>

#Parameter: b\_launchNETCA

#Type: Boolean

b\_launchNETCA=<Value Unspecified>

#Parameter: b\_javavmIsInstalling

#Type: Boolean

b\_javavmIsInstalling=<Value Unspecified>

#Parameter: b\_cmanIsInstalling

#Type: Boolean

b\_cmanIsInstalling=<Value Unspecified>

#Parameter: b\_anoIsInstalling

#Type: Boolean

b\_anoIsInstalling=<Value Unspecified>

#Parameter: s\_netCAInstalledProducts

#Type: String

s\_netCAInstalledProducts=<Value Unspecified>

[oracle.networking.netca\_9.2.0.1.0]

#Parameter: OPTIONAL\_CONFIG\_TOOLS

#Type: StringList

#Description: List of Optional Config tools that needs to be launched.

#The following choices are available. The value should contain only one of these choices.

#The choices are of the form Internal Name : External name. Please use the internal name while specifying the value.

```

# netca, : Oracle Net Configuration Assistant
#Example : OPTIONAL_CONFIG_TOOLS = {"netca"}
OPTIONAL_CONFIG_TOOLS=<Value Unspecified>

#Parameter: s_responseFileName
#Type: String
s_responseFileName=<Value Unspecified>

#Parameter: s_netCAInstalledProtocols
#Type: String
s_netCAInstalledProtocols=<Value Unspecified>

#Parameter: s_netCAInstalledProducts
#Type: String
s_netCAInstalledProducts=<Value Unspecified>

#Parameter: s_cfgtyperet
#Type: String
s_cfgtyperet=<Value Unspecified>

#Parameter: b_launchNETCA
#Type: Boolean
b_launchNETCA=<Value Unspecified>

[oracle.oid.tools_9.2.0.1.0]
#Parameter: s_OPSSelectedNodes
#Type: String
s_OPSSelectedNodes=<Value Unspecified>

[oracle.rdbms.common_schema_9.2.0.1.0]
#Parameter: s_bundleName
#Type: String
s_bundleName=<Value Unspecified>

[oracle.options.ops.opzca_9.2.0.1.0]
#Parameter: OPTIONAL_CONFIG_TOOLS
#Type: StringList
#Description: List of Optional Config tools that needs to be launched.
#The following choices are available. The value should contain only one of
these choices.
#The choices are of the form Internal Name : External name. Please use the
internal name while specifying the value.
# clustca, : Oracle Cluster Configuration Assistant
#Example : OPTIONAL_CONFIG_TOOLS = {"clustca"}
OPTIONAL_CONFIG_TOOLS=<Value Unspecified>

```

```

#Parameter: s_serverInstallType
#Type: String
s_serverInstallType=<Value Unspecified>

#Parameter: s_cfgtyperet
#Type: String
s_cfgtyperet=<Value Unspecified>

[oracle.webdb.modplsql_3.0.9.8.3b]
#Parameter: s_oracleApacheConfigFile
#Type: String
#Description: The Apache config file oracle_apache.conf will be passed with
this variable.
s_oracleApacheConfigFile=<Value Unspecified>

#Parameter: plsql_cache_dir
#Type: String
#Description: This is the location of the cache directory.
plsql_cache_dir=<Value Unspecified>

#Parameter: cookie_cache_dir
#Type: String
#Description: This variable holds the value of the cookie cache directory.
cookie_cache_dir=<Value Unspecified>

#Parameter: APACHE_HOME
#Type: String
#Description: The directory location where Apache is installed.
APACHE_HOME=<Value Unspecified>

[oracle.options.ops.pfs_9.2.0.1.0]
#Parameter: s_OPSSelectedNodes
#Type: String
s_OPSSelectedNodes=<Value Unspecified>

[oracle.rsf_9.2.0.1.0]
#Parameter: s_serverInstallType
#Type: String
s_serverInstallType=<Value Unspecified>

#Parameter: s_bundleName
#Type: String
s_bundleName=<Value Unspecified>

[oracle.rsf.rdbms_rsf_9.2.0.1.0]

```

#Parameter: s\_bundleName  
#Type: String  
s\_bundleName=<Value Unspecified>

[oracle.rsfxdk\_rsf\_9.2.0.1.0]  
#Parameter: s\_bundleName  
#Type: String  
s\_bundleName=<Value Unspecified>

[oracle.install.instcommon\_9.2.0.1.0]  
#Parameter: s\_OPSSelectedNodes  
#Type: String  
s\_OPSSelectedNodes=<Value Unspecified>

[oracle.rsfnlsrtl\_rsf\_9.2.0.1.0]  
#Parameter: s\_serverInstallType  
#Type: String  
s\_serverInstallType=<Value Unspecified>

[oracle.apache.jserv\_1.1.0.0.0g]  
#Parameter: s\_soapPort  
#Type: String  
#Description: The port that soap/JServ listens to  
s\_soapPort=<Value Unspecified>

#Parameter: s\_jservPort  
#Type: String  
#Description: The port Apache JServ listens to  
s\_jservPort=<Value Unspecified>

#Parameter: s\_LANGUAGE\_TERRITORY  
#Type: String  
#Description: Substring of n\_NLSLANG  
s\_LANGUAGE\_TERRITORY=<Value Required>

#Parameter: b\_autoPortDetect  
#Type: Boolean  
#Description: Set to true by default, set to false for silent install  
b\_autoPortDetect=<Value Unspecified>

#Parameter: s\_topDir  
#Type: String  
#Description: APACHE\_TOP directory for Apache Web Server  
s\_topDir=<Value Unspecified>

```

#Parameter: s_jvm
#Type: String
#Description: Path to the Java Virtual Machine
s_jvm=<Value Unspecified>

#Parameter: APACHE_HOME
#Type: String
#Description: location of apache home
APACHE_HOME=<Value Unspecified>

#Parameter: s_oracleApacheConfigFile
#Type: String
#Description: location of the oracle_apache.conf file
s_oracleApacheConfigFile=<Value Unspecified>

#Parameter: s_oracleJservPropertiesFile
#Type: String
#Description: Location of the jserv.properties file
s_oracleJservPropertiesFile=<Value Unspecified>

[oracle.apache.jsdk_2.0.0.0.0d]
#Parameter: s_topDir
#Type: String
#Description: APACHE_TOP directory for Apache Web Server
s_topDir=<Value Unspecified>

[oracle.rsfn.net_rsf_9.2.0.1.0]
#Parameter: s_bundleName
#Type: String
s_bundleName=<Value Unspecified>

[oracle.ocs4j_2.1.0.0.0a]
#Parameter: s_cachePort
#Type: String
#Description: Port to run local cache service on by default
s_cachePort=<Value Unspecified>

[oracle.rdbms.aqapi_9.2.0.1.0]
#Parameter: s_OPSSelectedNodes
#Type: String
s_OPSSelectedNodes=<Value Unspecified>

[oracle.apache.apache_1.3.22.0.0a]
#Parameter: OPTIONAL_CONFIG_TOOLS

```

```
#Type: StringList
#Description: List of Optional Config tools that needs to be launched.
#The following choices are available. The value should contain only one of
these choices.
#The choices are of the form Internal Name : External name. Please use the
internal name while specifying the value.
#  configtool3, : Starting HTTP Server
#  configtool8, : Starting Oracle HTTP service
#Example : OPTIONAL_CONFIG_TOOLS = {"configtool3"}
OPTIONAL_CONFIG_TOOLS=<Value Unspecified>

#Parameter: sl_OHs
#Type: StringList
#Description: Holds oracle homes on machine
sl_OHs=<Value Unspecified>

#Parameter: s_apacheServerAppendText
#Type: String
#Description: This variable holds a concatenation of port configurations
for apache server
s_apacheServerAppendText=<Value Required>

#Parameter: s_apachePortSSL
#Type: String
#Description: Port number that apache uses for SSL port when starting in
SSL mode. Default 443
s_apachePortSSL=<Value Unspecified>

#Parameter: s_apachePortNonSSL
#Type: String
#Description: Port number that apache uses when starting in non-ssl mode
s_apachePortNonSSL=<Value Unspecified>

#Parameter: s_apachePort
#Type: String
#Description: Port number that apache uses when starting in ssl mode
s_apachePort=<Value Unspecified>

#Parameter: s_NLSLANG
#Type: String
#Description: String containing NLSLANG info
s_NLSLANG=<Value Required>

#Parameter: b_autoStartApache
#Type: Boolean
#Description: pass in true to have apache autostart when not using top
level component
b_autoStartApache=<Value Unspecified>
```

#Parameter: b\_autoPortDetect  
#Type: Boolean  
#Description: True by default, this variable should be set to false for silent installs.  
b\_autoPortDetect=<Value Unspecified>

#Parameter: s\_topDir  
#Type: String  
#Description: top level directory for Apache Web Server  
s\_topDir=<Value Unspecified>

#Parameter: ServerRoot  
#Type: String  
#Description: root directory of apache server  
ServerRoot=<Value Unspecified>

#Parameter: APACHE\_HOME  
#Type: String  
#Description: Apache location off of ORACLE\_HOME  
APACHE\_HOME=<Value Unspecified>

[oracle.jdk\_1.3.1.0.1a]  
#Parameter: s\_jdkVersion  
#Type: String  
#Description: String containing version information  
s\_jdkVersion=<Value Unspecified>

#Parameter: isNT  
#Type: Boolean  
#Description: Returns true if OS is WinNT  
isNT=<Value Unspecified>

#Parameter: JRE\_LIBHOME  
#Type: String  
#Description: Entry point for the jre lib directory  
JRE\_LIBHOME=<Value Unspecified>

#Parameter: JRE\_LIB  
#Type: String  
#Description: Returns entry point of jre/lib  
JRE\_LIB=<Value Unspecified>

#Parameter: JRE\_BIN  
#Type: String  
#Description: Entry point at jre/bin  
JRE\_BIN=<Value Unspecified>

#Parameter: JDK\_HOME

#Type: String  
#Description: This is the top level directory where the jdk lives.  
JDK\_HOME=/oracle/IBMJava2-s390-131

[oracle.java.j2ee.core\_9.2.0.1.0]  
#Parameter: JDK\_HOME  
#Type: String  
#Description: This is the top level directory where the JDK lives.  
JDK\_HOME=<Value Unspecified>

[oracle.rsfs.sf\_rsf\_9.2.0.1.0]  
#Parameter: s\_bundleName  
#Type: String  
s\_bundleName=<Value Unspecified>

[oracle.swd.oui\_2.2.0.11.0]  
#Parameter: PROD\_HOME  
#Type: String  
#Description: Complete path where the product needs to be installed.  
#Example : PROD\_HOME = "C:\ProductName"  
PROD\_HOME=<Value Unspecified>

#Parameter: InstLocProp  
#Type: String  
InstLocProp=<Value Unspecified>

[oracle.swd.jre\_1.1.8.1]  
#Parameter: PROD\_HOME  
#Type: String  
#Description: Complete path where the product needs to be installed.  
#Example : PROD\_HOME = "C:\ProductName"  
PROD\_HOME=<Value Unspecified>

#Parameter: s\_OPSSelectdNodes  
#Type: String  
#Description: s\_OPSSelectdNodes\_DESC\_ALL  
s\_OPSSelectdNodes=<Value Unspecified>

[oracle.swd.jre\_1.3.1.0.0]  
#Parameter: PROD\_HOME  
#Type: String  
#Description: Complete path where the product needs to be installed.  
#Example : PROD\_HOME = "C:\ProductName"  
PROD\_HOME=<Value Unspecified>

```
#Parameter: s_OPSSelectedNodes
#Type: String
#Description: This variable recives the OPS selected nodes value.
s_OPSSelectedNodes=<Value Unspecified>
```

```
[oracle.doc.unixdoc_9.2.0.1.0]
#Parameter: s_OPSSelectedNodes
#Type: String
s_OPSSelectedNodes=<Value Unspecified>
```

```
[oracle.swd.oui_2.2.0.12.0]
#Parameter: PROD_HOME
#Type: String
#Description: Complete path where the product needs to be installed.
#Example : PROD_HOME = "C:\ProductName"
PROD_HOME=<Value Unspecified>
```

```
#Parameter: InstLocProp
#Type: String
InstLocProp=<Value Unspecified>
```

# Abbreviations and acronyms

<b>CP</b>	VM Control Program
<b>DASD</b>	Direct Access Storage Device
<b>FCON</b>	Fibre connection
<b>FTP</b>	File transfer protocol
<b>GID</b>	global ID
<b>IBM</b>	International Business Machines Corporation
<b>ITSO</b>	International Technical Support Organization
<b>IUCV</b>	Inter-User Communication Vehicle
<b>LPAR</b>	Logical Partition
<b>LVM</b>	Logical Volume Manager
<b>NIC</b>	Network Interface Card
<b>OEM</b>	Oracle Enterprise Manager
<b>OSA</b>	Open System Adapter
<b>OTN</b>	Oracle technical network
<b>OUI</b>	Oracle Universal Installer
<b>PM</b>	Performance Monitor
<b>PRF</b>	Performance Reporting Facility
<b>RAC</b>	Oracle Real Application Clusters
<b>RMAN</b>	Recovery Manager
<b>RMF</b>	Report Monitor Facility
<b>RMFPMS</b>	
<b>RTM</b>	VM Real Time Monitor
<b>VM</b>	virtual machine
<b>VMPRF</b>	VM Performance Reporting Facility
<b>YAST</b>	yet another system tool



# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

## IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 250.

- ▶ *Linux for IBM @server zSeries and S/390: Distributions*, SG24-6264
- ▶ *Linux for S/390*, SG24-4987
- ▶ *Linux on IBM @server zSeries and S/390: ISP/ASP Solutions*, SG24-6299
- ▶ *Building Linux Systems under IBM VM*, REDP0120
- ▶ *Linux on zSeries and S/390: Systems Management*, SG24-6820 (in review)
- ▶ *Large Scale Linux Deployment*, SG24-6824

## Other resources

These Oracle publications are also relevant as further information sources:

- ▶ *Oracle9i Installation Guide Release 2 (9.2.0.1.0), Part No. A96167-01*
- ▶ *Oracle9i Administrator's Reference Release 2 (9.2.0.1.0), Part No. A97297-01*
- ▶ *Oracle9i Release Notes, Release 2 (9.2.0.1.0) for Linux/390 Systems (31-Bit), August 2002, Part No. A97686-02*
- ▶ *z/VM V4R3.0 CP Planning and Administration*, SC24-6043
- ▶ *Hardware Management Console Operations Guide*, SC28-6815
- ▶ *z/VM V4R3.0 Performance Reporting Facility Function Level 410*, SC24-6027
- ▶ *z/VM V4R3.0 RealTime Monitor Function Level 410*, SC24-6028

## Referenced Web sites

These Web sites are also relevant as further information sources:

- ▶ Oracle Technical Network  
<http://www.otn.oracle.com>
- ▶ Oracle Homepage  
<http://www.oracle.com>

## How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

### IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

# Index

## Symbols

.rhosts file 156

## Numerics

64-bit architecture 2

## A

Apache Web server 178

AVGPROC 133

## B

backup and recovery tool 103

BMC PATROL software 206

## C

Central Processor. See CP

channel-to-channel. See CTC

cloning 14, 16

cloning Linux 83

CMOS hardware 188

compiling invalid objects 100

CP 15

CP commands

INDICATE 133

INDICATE USER 136

CP INDICATE Command 190

CP Monitor Facility 192

CPU utilization 133

CTC 19

CyberTrust 7

## D

Data Mining 7

Database Configuration Assistant 58

DBCA 149, 158, 176

dbca 58

dedicated CPs 22

dispatch list

definition 133

dormant list

definition 132

## E

eligible list

definition 132

engine 15

Entrust 7

ESALPS 208

Exceed software 197

export 88

## F

FCON/ESA. 195

## G

GiD 29

Guest Lan 16

## H

Hardware Management Console. See HMC

HiperSockets 16

hipersockets 154

HMC 13, 22

Power-on Reset 13

HMC operations

LPAR image profile 13

hosts.equiv file 155

HTTP server 178

## I

IBM WebSphere

Administrative Console 174

IBM Websphere

Administrative Repository 174

Administrative Server 174

Application Server 174

Repository Database 177

repository database 173

IFL 15, 22

image profile 13, 22

import 88

import log 101

import parameter file 100

import utility 100

import.sh script 100  
INDICATE command 133  
INDICATE USER 136  
inetd.conf file 155  
init.ora parameter 90  
Integrated Facility for Linux. See IFL  
Inter-User Communications Vehicle. See IUCV  
invalid objects 100  
iostat 148  
IUCV 19

## L

Label Security 7  
level of SuSE 29  
Linux configuration  
  cloning 14, 16  
Linux distributions 8  
Linux variables 29  
listener 176  
Logical Partition. See LPAR  
Logical Volume Manager 15  
Logical Volume. See LVM  
logical volumes 23  
LPAR 12, 21  
  activation 13  
  dedicated CPs 22  
  dispatcher 22, 25  
  image profile 22  
  shared CPs 22  
  weight 21, 25  
LVM 22  
  logical volumes 23  
  physical volumes 23  
  striping 22–23  
  volume groups 23

## M

migrate 87

## N

netstat 148  
Network File System. See NFS  
NFS 25

## O

object-code-only. See OCO  
OCO 15

OLAP 7  
Oracle Cluster Manager 149, 159  
Oracle Collaboration Suite 8  
Oracle e-business Suite 8  
Oracle Management Server (OMS) 7  
Oracle Technology Network 187  
Oracle Universal Installer 160  
Oracle9i Application Server 8  
OracleText 7  
oracm process 168  
OSA/Express 15, 18  
overcommitment 134–135  
overcommitment of main storage 14

## P

Partitioning 7  
performance 187  
Performance Reporting Facility (VMPRF) 193  
perfstat 144  
physical volumes 23  
Power-on Reset 13  
PR/SM 12  
private nodes 163  
Pro\*COBOL 7  
Processor Resource/Systems Manager. See PR/SM  
public nodes 162

## Q

QDIO 15, 18  
Queued Direct Input Output. See QDIO  
Quorum disk 165

## R

RAC 26  
raw devices 156  
Real Application Clusters 7, 149  
Real Application Clusters. See RAC  
Real Time Monitor (RTM) 194  
RealTime Monitor. See RTM  
Recovery Manager 103  
Red Hat 187  
Redbooks Web site 250  
  Contact us xiii  
remote copy 155  
remote shell 155  
repository database 173, 177

response file 217  
RMAN 103  
    ARCHIVELOG mode 104  
    backup command 112  
    CONFIGURE command 111  
    LIST command 116  
    recovery catalog 110  
    repository catalog 110  
    RESTORE/RECOVER command 114  
RMF PM 195, 207  
RTM 136

## S

SAR 205  
server consolidation 5  
server farms 6  
shared CPs 22  
sharing binaries 76  
SIZE390 21  
sizing 187  
sizing workloads 9  
Spatial 7  
SQL 88  
SQL\*Loader 88  
SRM 135  
SRM queues 132  
Statspac 142  
STORBUF 135  
striping 22–23  
SuSE 187  
SuSE 2.4.7 29

## T

TCP/IP 77, 154  
tools  
    export 88  
    import 88  
    IOSTAT 205  
    Linux Performance Measurement 201  
    RMF PM 207  
    SQL 88  
    SQL\*Loader 88  
    TOP 204  
    VM Performance Measurement 189  
    VMSTAT 202  
    XOSVIEW 203  
TOP 204

## U

upgrade 87  
Useful Linux Information 211

## V

VDISK in memory 137  
Verisign 7  
virtual machine 14  
VM  
    Accounting Records 191  
    CP Monitor Facility 192  
    FCON/ESA. 195  
    Performance Reporting Facility (VMPRF). 193  
    Real Time Monitor (RTM) 194  
VMSTAT 144, 202  
vmstat 142  
VNC software 197  
volume groups 23

## W

Watchdog 164  
watchdogd daemon 168  
WebSphere 173  
weight 21  
working set 137

## X

XOSVIEW 203  
xWindows Interface 197

## Z

z/VM commands  
    SET SRM 135  
z/VM Control Program (CP) 190  
z/VM Guest LAN 16  
z/VM SRM  
    queues 132  
z/VM system resources manager. see SRM





## Experiences with Oracle for Linux on zSeries

(0.2" spine)  
0.17" x 0.473"  
90 x 249 pages







**Redbooks**

# Experiences with Oracle for Linux on zSeries

## **Experiences installing Oracle9i**

This IBM Redbook describes experiences gained while installing and testing Oracle9i for Linux on zSeries, such as:

## **Options for implementing Oracle9i for Linux on zSeries**

- Setting up the development systems at Oracle for the Linux on zSeries environment
- Installing the Oracle9i instances for Linux/390 on zSeries
- Performing basic monitoring and tuning exercises

## **Managing Oracle9i on zSeries**

The redbook is based on real-world experiences gained by team members and technical professionals during development and early testing of this product at:

- The IBM Oracle EMEA Joint Solution Center, Montpellier, France
- The IBM Oracle International Competency Center, San Mateo, California
- Early customer installations
- Development systems at Oracle in Redwood Shores, California

This book will be of use to those customers who are using Oracle9i for Linux/390 on zSeries for the first time.

## **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](http://ibm.com/redbooks)