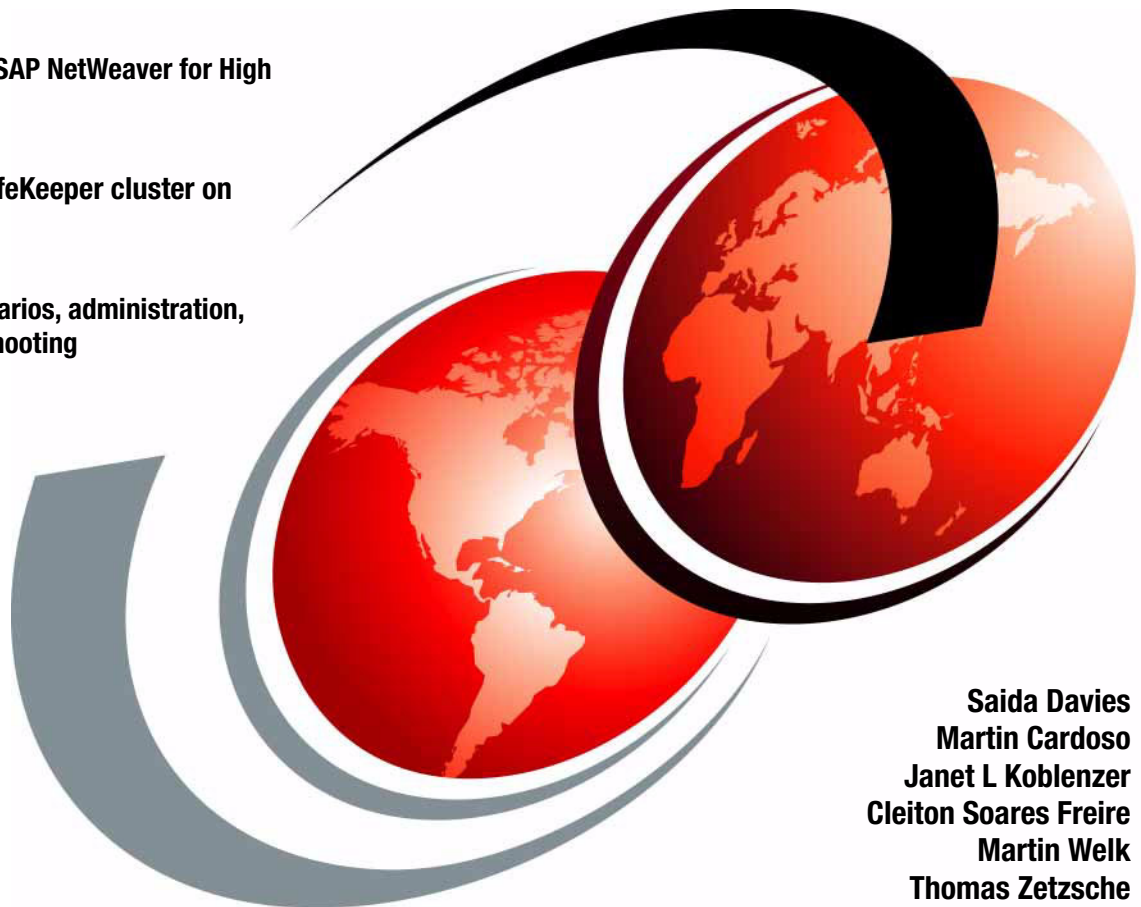


Building High Availability with SteelEye LifeKeeper for SAP NetWeaver on SUSE Linux Enterprise Server

Architecting SAP NetWeaver for High Availability

Building a LifeKeeper cluster on Linux

Failover scenarios, administration, and troubleshooting



Saida Davies
Martin Cardoso
Janet L Koblenzer
Cleiton Soares Freire
Martin Welk
Thomas Zetzsche



International Technical Support Organization

**Building High Availability with SteelEye
LifeKeeper for SAP NetWeaver on
SUSE Linux Enterprise Server**

July 2008

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

First Edition (July 2008)

Version 10, Service Pack 1 of Novell SUSE Linux Enterprise Server

Version 6, Release 2, Modification 1 of SteelEye LifeKeeper

Version 9, release 5 of IBM DB2

**Version 7, Release 0, Support Release 2 of SAP NetWeaver 7.0
(previously also known as 2004s)**

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	ix
Examples	xiii
Tables	xv
Notices	xvii
Trademarks	xviii
Preface	xix
The team that wrote this book	xx
Become a published author	xxiii
Comments welcome	xxiv
Chapter 1. High availability overview	1
1.1 The scope of this book	2
1.2 The goal of this book	2
1.3 What is covered in this book	2
1.4 What is not covered in this book	3
1.5 Assumptions	3
1.6 What is high availability?	3
1.7 High availability achieved	4
1.8 Cluster technologies	5
1.9 High availability maintenance	6
1.10 High availability definitions	6
1.10.1 Degrees of availability	6
1.10.2 Types of outages	7
Chapter 2. Introduction to SAP NetWeaver, Novell SUSE Linux Enterprise Server, and SteelEye LifeKeeper	9
2.1 SAP NetWeaver	10
2.1.1 The SAP NetWeaver technology component	10
2.1.2 Client/server configuration for SAP systems	12
2.1.3 SAP NetWeaver and high availability	14
2.2 SUSE Linux Enterprise Server	15
2.2.1 Historical background	15
2.2.2 SUSE Linux Enterprise Server and SAP	15
2.3 LifeKeeper for Linux	16
2.3.1 LifeKeeper for Linux components	16

Chapter 3. High availability architectural considerations	19
3.1 Server	20
3.1.1 Power supply	22
3.1.2 Memory	22
3.1.3 Ethernet network adapters	23
3.1.4 Storage Area Network adapters	23
3.1.5 Central processing unit	23
3.2 Base software	23
3.2.1 Cluster software	24
3.2.2 Base operating system software	27
3.3 Network	28
3.3.1 MAC address failover	31
3.3.2 IP address failover	31
3.3.3 Redundant network switch architecture	32
3.4 Storage	32
3.4.1 Direct Attached Storage (DAS)	33
3.4.2 Network Attached Storage (NAS)	33
3.4.3 Storage Area Network (SAN)	34
3.4.4 Multipath Storage Area Network connection	36
3.4.5 Replication	38
3.5 Database	40
3.5.1 Replication techniques	40
3.5.2 Cluster techniques	41
3.5.3 Additional considerations	45
3.6 SAP NetWeaver components	45
3.6.1 Central instance	46
3.6.2 Central services	46
3.6.3 SAP Central file systems	47
3.6.4 Network File System	47
3.6.5 Application server	48
3.7 SAP NetWeaver Single Points of Failure	48
3.7.1 Failure of the enqueue server	50
3.7.2 Failure of the database instance	51
3.8 SAP NetWeaver in cluster configurations	52
3.8.1 Active/Passive mode	53
3.8.2 Active/Active mode	54
Chapter 4. High availability topologies	57
4.1 Servers	59
4.1.1 Power supply	59
4.1.2 Random access memory	59
4.1.3 Hard disks	60
4.1.4 Local Area Network interfaces	60

4.1.5 Storage Area Network interfaces	61
4.1.6 Test environment	61
4.2 Network	65
4.3 Base operating system	65
4.3.1 Linux operating system	65
4.4 Cluster software	74
4.5 Storage	76
4.5.1 Storage scenario	76
4.5.2 Storage layout	78
4.6 IBM DB2	80
4.7 NetWeaver	81
4.7.1 NetWeaver 7.0 components	81
4.7.2 Protecting the points of failure	84
4.7.3 Database connectivity	87
4.7.4 SAP file systems	87
Chapter 5. High availability implementation	89
5.1 Prerequisites	90
5.1.1 Hardware	90
5.1.2 Compatibility	92
5.1.3 Software levels and requirements	93
5.1.4 Firmware	99
5.2 Base operating system installation	99
5.2.1 Naming	100
5.2.2 Installation	100
5.3 Shared storage	122
5.3.1 Host bus adapters	122
5.3.2 Configuring multipath connectivity	125
5.3.3 Configuring host-based mirroring	127
5.4 DB2 Linux, UNIX, and Windows Enterprise Server installation	129
5.4.1 Software installation	130
5.4.2 Configuration steps for DB2 on the standby server	134
5.4.3 DB2 Instance and database creation	135
5.4.4 Configuring DB2 settings after SAPinst	135
5.5 SAP NetWeaver installation	137
5.5.1 Media list	137
5.5.2 Steps for installation	138
5.5.3 SAP Central Services installation	140
5.5.4 Database installation	142
5.5.5 Central Instance installation	143
5.5.6 Installation of the enqueue replication server	144
5.5.7 Application Server installation	150

5.6 LifeKeeper cluster software installation	151
5.6.1 LifeKeeper Support CD installation	152
5.6.2 LifeKeeper software installation	154
5.7 Cluster configuration	155
5.7.1 Starting LifeKeeper	155
5.7.2 Creating a cluster configuration	159
5.8 Creating resources and hierarchies to protect applications	165
5.8.1 Creating file system resources	166
5.8.2 Creating IP resources	172
5.8.3 Creating database resource	176
5.8.4 Creating dependences	179
5.8.5 Creating SAP resources	182
Chapter 6. Testing and failover scenarios	197
6.1 Test methodology	198
6.1.1 Test steps	198
6.2 Failover scenarios	203
6.2.1 Failure of the active server	204
6.2.2 Failure of the standby server.	206
6.2.3 Failure of the ABAP central services.	209
6.2.4 Failure of the Java central services.	213
6.2.5 Failure of the central and application instances	214
6.2.6 Failure of the database system	217
6.2.7 Failure of the NFS Server	221
6.2.8 Planned outages	222
Chapter 7. Administering the cluster	223
7.1 Base operating system	224
7.1.1 Redundant Local Area Network connection	224
7.1.2 Redundant Storage Area Network connection	225
7.1.3 Mirroring data across storage sub-systems	226
7.1.4 Enhancing a file system on the shared storage	234
7.1.5 Frequent file system checks	235
7.2 SteelEye LifeKeeper administration	238
7.2.1 LifeKeeper services.	238
7.2.2 LifeKeeper Graphical User Interface.	240
7.2.3 Optional configuration tasks	243
7.2.4 Housekeeping	244
7.3 Customizing of LifeKeeper parameters	255
7.3.1 Changing global operational parameters	255
7.3.2 Changing LifeKeeper configuration values	256

7.4 Maintenance during uptime	257
7.5 Backup and restore	260
Chapter 8. Troubleshooting	263
8.1 Installation	264
8.2 DB2	266
8.3 SAP NetWeaver	267
8.4 LifeKeeper	269
Appendix A. Additional material	271
Locating the Web material	271
Abbreviations and acronyms	273
Related publications	275
IBM Redbooks	275
Online resources	275
How to get Redbooks	278
Help from IBM	278
Index	279

Figures

2-1	SAP NetWeaver four main components	11
2-2	SAP NetWeaver layers	14
2-3	LifeKeeper for Linux components	18
3-1	SteelEye LifeKeeper hierarchies	26
3-2	The Linux enabled IBM product family	27
3-3	Redundant network connection.	29
3-4	Open Systems Interconnection (OSI) Basic Reference Model	30
3-5	SAN Components with two arrays	35
3-6	Redundant Storage Area Network connectivity	37
3-7	I/O timing in Metro and Global Mirroring	39
3-8	Clustering databases compared to replication	42
3-9	Hybrid HA solution	43
3-10	Failover or HA database cluster	44
3-11	SPOFs within the SAP components	49
3-12	Enqueue server high availability	51
3-13	Cascading cluster configuration	52
3-14	Active/Passive mode.	53
3-15	Active/Active mode	54
4-1	IBM Blade Center	62
4-2	IBM Blade server HS 21	62
4-3	A single management module for the IBM Blade Center	63
4-4	A view to the Blade Center Management Module.	64
4-5	Example for disk mirroring with the Software RAID driver module	69
4-6	Example for a volume group with the Linux Logical Volume Manager	71
4-7	Overview of the storage related Linux driver stack	73
4-8	Storage topology in the test scenario	77
4-9	Storage configuration for the test scenario	79
4-10	Logical view to the application components	82
4-11	Distribution of services in switchover groups	84
4-12	Logical view of a fail-over situation	86
5-1	Sizing methods and relation with the project phase	91
5-2	Operating system installation; graphical boot menu	101
5-3	Language selection	103
5-4	Time zone setting	104
5-5	Create custom partition setup	105
5-6	Custom partitioning	105
5-7	Disk partitioning.	106
5-8	Create a volume group	107

5-9 Overall disk layout	108
5-10 Software selection	109
5-11 Installation settings	110
5-12 Network setup method	111
5-13 Network card configuration overview	112
5-14 Network address setup (address)	113
5-15 Hostname and name server configuration	114
5-16 Routing configuration	115
5-17 Network address setup (general)	116
5-18 Network time protocol configuration	119
5-19 Select the installation directory window	131
5-20 Install the IBM Tivoli SA MP component window	132
5-21 Set up a DB2 instance window	133
5-22 High availability installation steps	138
5-23 Installing central services for ABAP and Java	141
5-24 Central Instance installation	144
5-25 Enqueue Replication Server topology	145
5-26 Directory structure for an enqueue replication instance	145
5-27 SAP hierarchy	149
5-28 SAP Application Server installation	151
5-29 Cluster Connect	160
5-30 Server context menu	161
5-31 Toolbar	161
5-32 Edit menu	161
5-33 Server context toolbar	162
5-34 List box Remote Server	162
5-35 Select box Device Type	163
5-36 List box of local IP addresses	163
5-37 Message box	164
5-38 Server icons indicating non redundant communication path	165
5-39 Server icons indicating redundant communication paths	165
5-40 Toolbar icon Create Resource Hierarchy	166
5-41 Selection of resource kit type	166
5-42 Select list for available file systems	168
5-43 Message during creation process	169
5-44 Success message	169
5-45 Pre extend checks	170
5-46 Hierarchy extended successful	171
5-47 Enter the IP address	172
5-48 Select or enter the network mask	173
5-49 Select or enter the Network Interface	174
5-50 create IP resource	175
5-51 Extend IP hierarchy	176

5-52	Dialog box Create Dependency	179
5-53	Create Dependency	180
5-54	Message box.....	180
5-55	Database resource	181
5-56	SAP hierarchy	185
5-57	Application Info Field.....	191
5-58	Resource Tag	192
5-59	SAP NetWeaver hierarchy including Central instance and SAP central services	193
5-60	Resource context	194
5-61	Properties window.....	195
5-62	Message box.....	195
5-63	SAP NetWeaver and DB2 database in an active/active configuration ..	196
6-1	ABAP SAP system RDB through the logon group	200
6-2	Checking availability of Java instance.....	201
6-3	ABAP enqueue server replication status.....	202
6-4	Java enqueue server replication status.....	202
6-5	Lock entries in SAP enqueue table	205
6-6	Active instances in se02	206
6-7	Connection to the D15 instance on se02	207
6-8	Lock entries in SAP RDB system	210
6-9	Java lock entries	214
6-10	Active Instances	215
6-11	Java Instances status	216
6-12	Remaining application after central instance shutdown	217
6-13	SAP RDB system log during database failure.....	220
7-1	Software RAID synchronization status in the LifeKeeper GUI	232
7-2	Running resource reconfiguration.....	235
7-3	In Service	247
7-4	Out of Service	248
7-5	Server Properties Panel	249
7-6	Choose View Logs	251
7-7	Log file types.....	252
7-8	LifeKeeper GUI with one resource Out of Service inside the hierarchy ..	259
7-9	Adding backup clients to the cluster hierarchy	261
8-1	Integrated File System (IFS) directory structure	268

Examples

4-1 Ethernet bonding configuration	66
5-1 VNC command	102
5-2 disk output.	106
5-3 Physical volumes	107
5-4 Logical volumes	108
5-5 Modifying /etc/sysconfig/windowmanager.	117
5-6 Modifying /etc/sysconfig/cron	118
5-7 Changes in /etc/ssh/sshd_config	120
5-8 Changes in /etc/ssh/ssh_config	121
5-9 Example of creating the /etc/ssh/ssh_known_hosts file	121
5-10 Content of /etc/ssh/ssh_known_hosts.	122
5-11 Overview about SCSI busses and devices	123
5-12 Example script showing devices grouped by WWN, show_equal_wwn	123
5-13 List of Linux SCSI disk devices grouped by WWN	124
5-14 Linux multipath configuration file, /etc/multipath.conf	125
5-15 Output of the multipath command	126
5-16 Creating Software RAID mirrors	127
5-17 Output of mdadm --query --detail /dev/md1	128
5-18 Configuration file /etc/mdadm.conf	128
5-19 Mdadm configuration.	129
5-20 DB2 registry variables	136
5-21 Setting variable for installation	140
5-22 Starting sapinst interface.	140
5-23 ImportMonitor.console.log file output	143
5-24 SAP copy list file	146
5-25 Enqueue server start profile	147
5-26 Replication enqueue instance profile	148
5-27 Parameter for SAP central services instance profiles	148
5-28 Change on DEFAULT.PFL profile file	149
5-29 Start message of LifeKeeper.	156
5-30 Start message of GUI server.	156
5-31 Entry in /etc/fstab	182
5-32 Soft links under /usr/sap/RDB/SYS.	182
5-33 Soft links under /usr/sap/RDB/SYS/exe	183
5-34 Template for restore script	186
5-35 Template for remove script	187
5-36 Template for quickCheck script.	188
5-37 Template for recovery script	189

6-1 Mounted file systems in the active server	198
6-2 Availability of the ABAP Central Services ASCS10	199
6-3 Availability of the Java Central Services SCS11	200
6-4 Checking enqueue replication status	201
6-5 Cluster management log from the failure to the recovery	208
6-6 Replication verification	210
6-7 Message server failure and recovery	211
6-8 Enqueue server replication status.	213
6-9 Java message server availability	214
6-10 DB2 threads running on server se01	218
6-11 Terminating NFS Server	221
7-1 Content of /proc/net/bonding/bond0	224
7-2 Status output of the multipath command.	225
7-3 Content of configuration file /etc/mdadm.conf	226
7-4 Content of /proc.	227
7-5 Example of mdadm detailed output.	228
7-6 Get device id from device path	228
7-7 Device entries in the /dev/mapper directory	229
7-8 Alias names from the Device-Mapper Multipaths I/O module	230
7-9 Setting one disk of a mirror to faulty state.	230
7-10 Sample output of /proc/mdstat with a failed device.	230
7-11 Removing and re-adding a failed drive	231
7-12 Content of /procM	231
7-13 Showing and setting the Software RAID speed limits.	233
7-14 Creating a snapshot for a logical volume	236
7-15 Watching snapshot usage.	237
7-16 Running a file system check on a snapshot	237
7-17 Remove snapshot and set current date	237
7-18 Main LifeKeeper processes.	239
7-19 Profile for LifeKeeper	243
7-20 Restore script for mdadm resource.	243
7-21 QuickCheck script for mdadm resource	243
7-22 Recovery script for mdadm resource	244
7-23 Remove script for mdadm resource	244
7-24 Output from the command lcdstatus -q	245
7-25 Output lksupport	254
8-1 Output from sapinst.log	264
8-2 Sample listing of DB2 products and features	267
8-3 output from lk_log	269

Tables

4-1	SAP RDB system instances of the test environment	83
4-2	Shared file systems.	87
5-1	SUSE Linux Enterprise Server kernel parameters for DB2.	96
5-2	SUSE Linux Enterprise Server packages requirement for DB2	96
5-3	Local file systems	97
5-4	Shared file systems.	98
5-5	TCP/IP address/host name mapping in the test environment.	100
5-6	Media list for the test environment	137
5-7	Variables used in the book test environment installation	139
5-8	Installation display options	139
5-9	File copies to ERS28 directory structure.	146
6-1	Failure scenarios.	203
6-2	Status of the active server before the failure.	204
6-3	Summary of the standby server failure simulation	206
6-4	Summary of ASCS10 instance test failure	209
6-5	Java central service test summary	213
6-6	Central instance failure simulation	215
6-7	Summary of the database failure simulation	218
6-8	NFS Server failure in the active server	221
6-9	Planned outage scenarios.	222
7-1	Status codes for resources	245

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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


COPYRIGHT LICENSE:

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

AIX®	IBM®	System x™
BladeCenter®	Informix®	System z™
Chipkill™	Parallel Sysplex®	Tivoli®
DB2 Universal Database™	Redbooks®	TotalStorage®
DB2®	Redbooks (logo)  ®	WebSphere®
DS4000™	ServeRAID™	X-Architecture®
DS6000™	System p™	z/OS®
DS8000™	System Storage™	

The following terms are trademarks of other companies:

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

AppArmor, Novell, SUSE, the Novell logo, and the N logo are registered trademarks of Novell, Inc. in the United States and other countries.

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

ABAP, SAP NetWeaver, SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

LifeKeeper, SteelEye Technology, SteelEye, and the SteelEye logo are registered trademarks of Steeleye Technology, Inc. Other brand and product names used herein are for identification purposes only and may be trademarks of their respective companies.

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

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

Intel Xeon, Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

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

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

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

Preface

Business processes based on the SAP® NetWeaver platform often require a high level of availability and fault tolerance.

Availability can be defined as the amount of time application services are accessible to the end user, and is measured by the percentage in time that the application is available to the user. The application is highly available when it gets closer to the difficult-to-achieve 99.999% threshold of availability, also known as the five 9s of availability.

The Novell® SUSE® Linux® Enterprise Server base operating system, SteelEye LifeKeeper for Linux cluster software, and SAP NetWeaver software provide capabilities that can be implemented to build a configuration that fulfills these requirements.

In this book, we discuss the concepts of high availability, provide an overview about the main components, and explain high availability configurations and implementation.

We cover these topics:

- ▶ Server hardware configuration
- ▶ Novell SUSE Linux Enterprise Server software installation
- ▶ SteelEye® LifeKeeper® cluster software installation
- ▶ Network topology and configuration
- ▶ Storage topology and configuration
- ▶ DB2® software installation
- ▶ SAP Netweaver software installation

We also discuss the integration, test, and operation of all these components.

Technical scenarios are illustrated and verified with aspects of troubleshooting, describing errors encountered in the scenarios. The scenarios focus on:

- ▶ High availability architectural considerations
- ▶ High availability topologies
- ▶ High availability implementation
- ▶ Testing
- ▶ Administering the cluster
- ▶ Troubleshooting

The team that wrote this book



From left: Cleiton, Martin, Saida, Thomas, Janet, Martin
Photo taken by Anja Welk

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

Saida Davies is a Project Leader for the International Technical Support Organization (ITSO) and has extensive experience in Information Technology. She has published several Redbooks® and Redpapers® on WebSphere® Business Integration, Web Services, and WebSphere Service Oriented Middleware using multiple platforms. Saida has experience in the architecture and design of WebSphere MQ solutions, extensive knowledge of z/OS® operating system, and a detailed working knowledge of both IBM® and Independent Software Vendors' operating system software. As a Senior IT Specialist, her responsibilities included the development of services for WebSphere MQ within the z/OS and Windows® platform. This covered the architecture, scope, design, project management and implementation of the software on stand-alone systems or on systems in a Parallel Sysplex® environment. She has received Bravo Awards for her project contributions. Saida has a degree in Computer Studies and her background includes z/OS systems programming. Saida supports Women in Technology activities and she contributes to and participates in their meetings.

Martin Cardoso is an IT Architect working in the SAP Tiger Team for Business Global Services in IBM Argentina. He joined IBM in 2002 from PwC Consulting, working as an SAP Basis Consultant and SAP Basis SME for the SSA region. Martin has more than ten years experience working with SAP technologies. His expertise include storage and related software, database technologies, SAP NetWeaver technologies, SAP capacity planning, SAP performance and tuning, SAP heterogeneous migrations, and he now specializes in designing of SAP infrastructure architectures. He is a co-author of the Redbooks publication, *Best Practices for SAP BI using DB2 9 for z/OS*, SG24-6489. Martin was certified as an SAP NetWeaver Technical Consultant in 2007.

Janet L Koblenzer joined IBM in 2000 and is a Senior IT Specialist in the IBM Systems and Technology Group (STG). She has over twenty years experience in the Information Technology field with a primary focus on database management. Her expertise includes both System z™ and open systems database platforms in systems management, performance tuning, and architectural design for high availability. Her particular interest lies in DB2 database technologies. Janet currently works for the System z Lab Services team based in Poughkeepsie, NY assisting customers in the deployment of database solutions. These range from Linux on System z consolidations, Business continuity and resilience architecture and ISV Packaged Solution implementations for DB2 Linux, UNIX®, and Windows (LUW), Oracle®, and DB2 z/OS. Janet holds a Masters degree in Business from Cleveland State University.

Cleiton Soares Freire is an IT Specialist working for IBM Global Services in Brazil. His main responsibility is migrating SAP systems from several vendor platforms to IBM AIX®. His additional responsibility is to provide Third Level Support to IBM customers across South America and United States. Cleiton is a certified Technology Consultant for the SAP NetWeaver platform and is a Bachelor of Science graduated from Faculty Of Technology, in Sao Paulo, Brazil.

Martin Welk is a systems engineer in IBM GTS ITS division. He started his IT career in the early 90's, building his experience in Internet technologies. Martin built ISP environments for a few years and joined IBM in 2000. He started with designing and building e-business infrastructures on different UNIX systems focusing on high availability. In 2003, he participated in migrating large SAP and Oracle based environments to SUN E15000 platform. Since 2004, he has worked primarily with Linux as a base operating system. He is currently working as an architect focusing on open source technology. Martin's role in the Enterprise Linux Services group is assisting customers to migrate to open source software based infrastructure, especially with IBM Tivoli® Storage Manager, IBM Tivoli System Automation for Multi-platforms, SteelEye LifeKeeper, Oracle RDBMS, Xen, VMware, Samba, and OpenLDAP. He is familiar with IBM System x™ and System p™ platforms, and IBM and non-IBM enterprise storage. Martin achieved a certification for SteelEye Life Keeper and the ITIL® foundation certificate in 2005, and a Red Hat Certified Engineer qualification in 2006.

Thomas Zetzsche a consultant for CC Computersysteme und Kommunikationstechnik GmbH. He started his IT career in 1994, building his expertise in SAP basic components. Thomas joined his current company fifteen years ago. Over the course of his career he has worked on various software products including LifeKeeper, SAP, Oracle, MaxDB, DB2, Linux, and Windows. In the last eight years his work has been to design, implement, and support high availability solutions for Linux and Windows. Thomas is a member of the SteelEye Competence and Support Center for Central and Eastern Europe, where his responsibility is assisting customers to implement high availability solutions based on SteelEye LifeKeeper in SAP and database environments. Thomas has participated in many successful projects integrating SteelEye LifeKeeper in SAP environments.

The ITSO would like to express its special thanks to the IBM SAP International Competence Center (ISICC) in Walldorf, Germany for hosting this project, providing the hardware, software and access to systems.

Sincere thanks to the following persons for supporting this project:

Ted Davis
OPS ISV Enablement
Novell Inc.
SUSE Linux Enterprise Server 10

Bonni-Jo B. Salazar
VP, Strategic Alliances
SteelEye

Antonio Palacin
Director of IBM SAP International Competence Center, IBM Sales & Distribution,
Software Sales, IBM Germany

Bernd Schoener, IBM/SAP Alliance Technology Executive
Senior IT Architect, IBM SAP International Competence Center
IBM Germany

Also many thanks for his assistance in the planning and facilitating of this residency to run at the ISICC Walldorf, Germany.

The Redbooks publication team would like to thank the following people for their guidance, assistance, and contributions to this project:

Paul Henter
Senior Certified Consultant - IBM System x SAP Solutions, IT Management
Consultant, IBM Sales & Distribution, Software Sales, IBM SAP International
Competence Center, IBM Germany

Michael Siegert
Senior IT Specialist WW Technical Enablement SAP on IBM System x Solutions,
IBM SAP International Competence Center, IBM Germany

Walter Orb
Server Specialist for SAP Solutions on System p, IBM Sales & Distribution,
Software Sales, IBM SAP International Competence Center, IBM Germany

Volker Nickel
Network Specialist, IBM Global Technology Services, TSS IGA Germany North /
West, IBM Germany

Gerd Jelinek
General Manager, CC Computersysteme und Kommunikationstechnik GmbH,
Dresden, Germany

Robert Heinzmann
High Availability Consultant for SteelEye, CC Computersysteme und
Kommunikationstechnik GmbH, Dresden, Germany

Dr. Mira Stranz
Pre-sale Manager for SteelEye Competence and Support Center,
CC Computersysteme und Kommunikationstechnik GmbH, Dresden, Germany

Pablo Martin Pessagno
Technical Solution Architect, IBM Global Technology Services, ITS SS&SS, IBM
Argentina

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

ibm.com/redbooks

- ▶ Send your comments in an e-mail to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

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



High availability overview

Many of today's corporations run their businesses on the SAP application suite. High availability of these applications to the user community is essential because downtime means lost sales and profits. The combination of SAP, Novell SUSE Linux Enterprise Server, and LifeKeeper are important elements to provide the required access to key business applications.

In this chapter, we cover the following topics:

- ▶ “The scope of this book”
- ▶ “The goal of this book”
- ▶ “What is covered in this book”
- ▶ “What is not covered in this book”
- ▶ “Assumptions”
- ▶ “What is high availability?”
- ▶ “High availability achieved”
- ▶ “Cluster technologies”
- ▶ “High availability maintenance”
- ▶ “High availability definitions”

1.1 The scope of this book

This book discusses the concepts of high availability and provides an overview of SAP NetWeaver, Novell SUSE Linux Enterprise Server, and SteelEye Lifekeeper. High availability architectural considerations, availability topologies, and high availability implementations are explained.

Installation and configuration of the core software are covered. These include installation of the base operating system, database, shared storage, cluster management software and the SAP NetWeaver framework.

Technical failover scenarios are illustrated and tested. Troubleshooting techniques with problem resolution are provided.

1.2 The goal of this book

The goal of this book is to provide a high availability technology solution to meet critical business requirements. This solution demonstrates a reliable and resilient high availability architecture for SAP NetWeaver infrastructure with LifeKeeper. The operating system is Novell SUSE Linux Enterprise Server running on IBM System x hardware. This book helps deliver required service levels in a cost effective way.

1.3 What is covered in this book

We provide a basic understanding of high availability concepts and design considerations, including technical detail on how high availability can be achieved and tested. Best practice methods, administration considerations, and troubleshooting guidelines are also discussed.

The book is organized in the following chapters:

- ▶ Chapter 1., “High availability overview”
- ▶ Chapter 2., “Introduction to SAP NetWeaver, Novell SUSE Linux Enterprise Server, and SteelEye LifeKeeper”
- ▶ Chapter 3., “High availability architectural considerations”
- ▶ Chapter 4., “High availability topologies”
- ▶ Chapter 5., “High availability implementation”
- ▶ Chapter 6., “Testing and failover scenarios”

- ▶ Chapter 7., “Administering the cluster”
- ▶ Chapter 8., “Troubleshooting”

1.4 What is not covered in this book

This book does not replace the installation guides, related notes, or information for Novell SUSE Linux Enterprise Server, SAP NetWeaver, or SteelEye LifeKeeper. Rather, it is a guide on how to build a high availability infrastructure, with helpful instructions on how to implement it.

This book has to be complemented with the latest notes from each product related to current versions and patches.

1.5 Assumptions

The first two chapters require no previous technical knowledge and provide an introduction to high availability, definitions, and an overview of the infrastructure.

The remaining chapters discuss installation, configuration, testing, administration, and troubleshooting of SteelEye LifeKeeper for SAP NetWeaver over the Novell SUSE Linux Enterprise Server high availability infrastructure. This book assumes good knowledge of system administration on Novell SUSE Linux Enterprise Server, technical SAP NetWeaver skills, and experience in SteelEye LifeKeeper or other cluster technologies.

1.6 What is high availability?

Availability can be defined as the amount of time application services are accessible to an end user and is measured by the percentage in time that the application is available to an end user. The application is considered highly available as it approaches the 99.9999% of availability commonly referred to as the five 9s of availability.

The Information technology (IT) department must ensure to achieve the business requirements by building a highly availability infrastructure that can be cost effective for the entire organization.

Availability of a system is the combination of availability of both components of solution, the infrastructure and application, such as:

- ▶ The hardware layer, such as servers and storage
- ▶ The operating system, such as Linux or Windows
- ▶ The base services, such as the data base
- ▶ The applications, such as SAP NetWeaver

Extended measurement includes the user network, the end user interface, and the user terminal, such as a personal computer.

In this book, high availability covers all of the foregoing levels. The end user access and related network components are not covered.

1.7 High availability achieved

Availability can be improved by eliminating single points of failure (SPOFs). There is a conceptual limit on this, and the data center on which the servers are located is a SPOF. To alleviate this, there are a number of different solutions available for disaster recovery. Disaster recovery refers to the capability to recover an entire data center to a different location. The topic of disaster recovery is beyond the scope of this book. Within the scope of this book, the term availability refers to availability within a single site.

It is easier to build a high availability solution as early as possible; starting from the design phase and using standard components are key for building a cost effective solution. Availability can be improved on an existing installation by minimizing single points of failure.

One method to improve the hardware availability is to implement clustering by using two or more computers or nodes for a common set of tasks. If one computer fails, then the others can take over the failed service. A second computer can be used as the redundant backup for the first computer. This design supports high availability.

Clustering can be used to increase the computing power of the entire computer installation. This also allows a system to be scalable. Adding more computers increases the power and hence the design can support more users. In this book, we use clustering for application availability with increasing processing power as a fortunate side effect.

Eliminating or masking single point of failure at the operating system, data base, or application level, usually takes more time and resources than putting the same task into the hardware layer. Statistically, there are four-fifths of unplanned outages for an installation versus one-fifth for the hardware layer.

1.8 Cluster technologies

A cluster can be defined as a collection of interconnected complete computers, or nodes, that appear on a network as a single unit.

The cluster is managed as a single system or operating entity. It is designed to:

- ▶ Tolerate component failures.
- ▶ Support the addition or subtraction of components in a way that is transparent to users.
- ▶ Accumulate resources for processing power.

Clustering becomes an important concept for both high availability and disaster recovery discussions. This book focuses on clustering to achieve high availability. There are two basic approaches for clustering:

- ▶ Hardware clustering: This requires specialized hardware. It usually involves a strong investment, both on the technical resources, for example, with special training and maintenance cost.
- ▶ Software clustering: This uses standard hardware and is a cost-effective solution for high availability.

SteelEye LifeKeeper is a clustering software application that ensures high availability of applications. LifeKeeper maintains the high availability of clustered systems by monitoring system and application health, maintaining client connectivity, and providing minimized downtime.

With LifeKeeper, hardware component or application faults are detected in advance of a full system failure through multiple fault-detection mechanisms.

LifeKeeper monitors clusters using intelligent processes and multiple heartbeats. By sending redundant signals between server nodes to determine system and application health, LifeKeeper confirms a system's status before taking action. This reduces the risk of a single point of failure and minimizes false failovers. LifeKeeper also limits unnecessary failovers by recovering failed applications, without a full failover to another server, if the hardware is still active.

In the event of an interruption in a server's availability, LifeKeeper automatically moves the protected resources and applications to another server in the cluster.

1.9 High availability maintenance

The other key factor is the operation and administration of the high availability environment. The statistical data shows that two-fifths of unplanned outages result from operator errors and unexpected user behavior.

To mitigate this, there are proactive monitoring tools that can enhance a clustered environment. SteelEye LifeKeeper provides an optional recovery kit for SAP that can monitor and automatically respond to SAP NetWeaver related failures. Thus, it eliminates the single points of failure and simplifies the SAP NetWeaver and cluster related configuration.

In a highly available infrastructure, best practice and ITIL processes need to be adhered to. There should be processes defined for both change and problem management. Detailed policies, validation, and testing procedures should be maintained throughout the lifecycle of the application.

1.10 High availability definitions

In this section the terms used to indicate various degrees of availability are defined. Two types of outages affecting availability are discussed that you need to be aware of.

1.10.1 Degrees of availability

The terms high availability, continuous operation, and continuous availability are generally used to express how available a system is. In the following sections, we define and discuss each of these terms.

High availability

High availability means being able to avoid unplanned outages by eliminating single points of failure. This is a reliability measure of hardware, operating system, and database manager software. Another measure of high availability is the ability to minimize the effect of an unplanned outage by masking the outage from the end users.

Continuous operation

Continuous operation means being able to avoid planned outages. For continuous operation, there must be ways of performing administrative work as well as hardware and software maintenance while the application remains available to the end users. This is accomplished by providing multiple servers and switching end users to an available server at times when one server is made unavailable.

It is important to note that a system running in continuous operation is not necessarily operating with high availability because the number of unplanned outages could be excessive.

Continuous availability

Continuous availability combines the characteristics of high availability and continuous operation to provide the ability to keep the SAP system running as close to 24x7x365 as possible. This is what most customers want to achieve.

1.10.2 Types of outages

Many corporations run their businesses on the SAP application suite. High availability of these applications to the user community is essential because downtime means lost sales and profits. Based on that, the availability of the SAP system is a critical business factor; therefore the highest level of availability must be provided. Customers must be aware of the types of outages and how to avoid them. Next we explain conceptual aspects of planned and unplanned outages.


Planned outage

Planned outages are deliberate and are scheduled at a convenient time. These involve activities such as:

- ▶ Database administration such as offline backup or offline reorganization
- ▶ Software maintenance of the operating system or database server
- ▶ Software upgrades of the operating system or database server
- ▶ Hardware installation or maintenance

Unplanned outage

Unplanned outages are unexpected outages that are caused by the failure of any SAP NetWeaver, Novell SUSE Linux Enterprise Server, LifeKeeper, and database system components. These include hardware failures, software issues, or people and process issues. We recommend including a procedure usually called *root cause analysis* to detect, document, and mitigate the origin of this outage.



Introduction to SAP NetWeaver, Novell SUSE Linux Enterprise Server, and SteelEye LifeKeeper

In this chapter we provide an overview of the main products discussed throughout this book:

- ▶ SAP NetWeaver
- ▶ SUSE Enterprise Server highlights
- ▶ SteelEye LifeKeeper for Linux

2.1 SAP NetWeaver

SAP NetWeaver is the foundation for SAP Solutions. It provides technical functions for all the business applications based on it.

SAP NetWeaver has an integrated runtime and development environment, providing enough infrastructure to enable customers to build and run their own applications. Since it was built using standard protocols, it can be easily integrated with IBM WebSphere and Microsoft®.Net.

2.1.1 The SAP NetWeaver technology component

SAP NetWeaver 7.0 is the latest available version of the of the NetWeaver framework. SAP NetWeaver framework is a set of applications based on the NetWeaver technology.

NetWeaver 7.0 is the same as NetWeaver 2004s renamed in the middle of 2007 in order to adapt to the new SAP naming conventions.

SAP NetWeaver has four main components:

- ▶ Application Platform
- ▶ Process Integration
- ▶ Information Integration
- ▶ People Integration

A graphical overview of the framework can be seen in Figure 2-1.

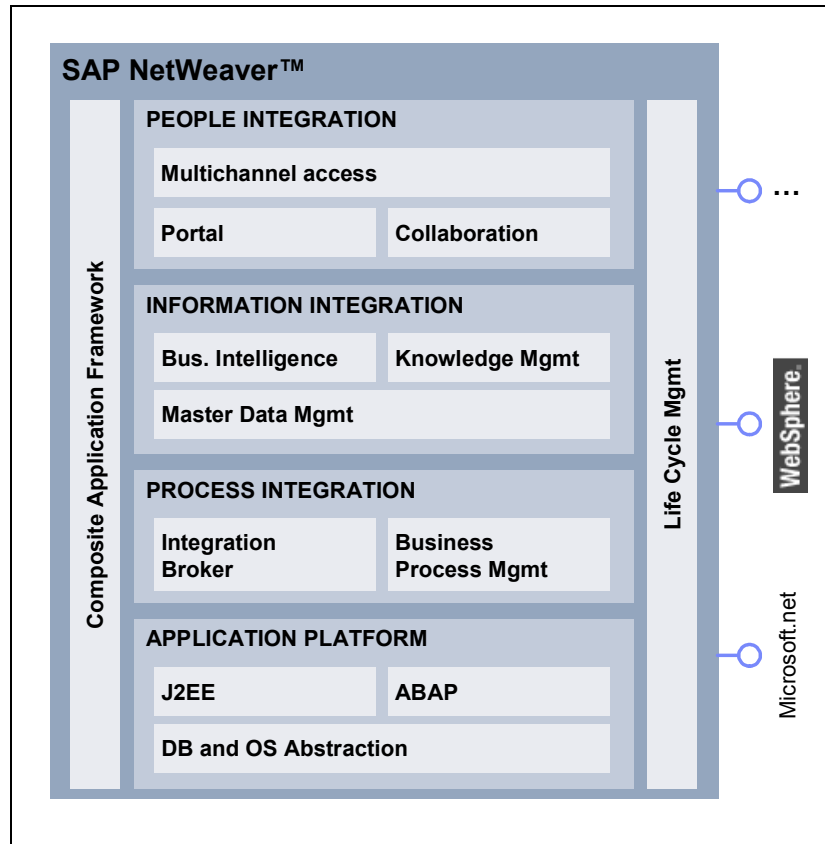


Figure 2-1 SAP NetWeaver four main components

Application Platform

The base of the SAP Application Platform component is the SAP Web Application Server, which supports several databases and operating system combinations. The availability matrix is accessible at:

<http://service.sap.com/pam>

SAP Web Application Server is a development of the previous application platform called SAP Basis.

SAP Application Server, besides the traditional ABAP™ development and runtime environment, now includes an implementation of the Java™ environment. The SAP J2EE™ engine, the Java environment, enables SAP NetWeaver to support Web services in an open environment.

Process Integration

The Process Integration component allows SAP NetWeaver to integrate easily, and with a great degree of reliability, to several different kinds of business programs.

The SAP Process Integration (SAP PI) is the command center to create an infrastructure of connections between SAP systems and several other systems.

Information Integration

The Information Integration component of SAP NetWeaver is composed of the following products:

- ▶ Master Data Management (MDM)
- ▶ Business Intelligence (BI)
- ▶ Knowledge Management (KM, a component of the Enterprise Portal).

With these tools, all business information can be centralized and structured in a single place, according to the business needs.

People Integration

All the information provided by the previous components can be displayed with this component.

People Integration provides a Web user interface for all information structured under the SAP NetWeaver.

The Enterprise Portal (EP), is the main component here, it can be used as a single point of access for all the business information.

2.1.2 Client/server configuration for SAP systems

An SAP system consists of three business application software layers:

- ▶ Presentation processes (Presentation layer)
- ▶ Application processes (Application layer)
- ▶ Database processes (Database layer)

Presentation processes: This layer is the front-end for user input and is passed to the next layer for processing. It is possible to attach several kinds of front-end to an SAP system, such as SAPGUI, Web-Browser, PDA, and other mobile devices.

Application processes: In the application layer, the execution of application programs are processed. This layer is the bridge between the user interface and the database.

Database processes: All data is stored only in the database layer. The database receives requests from the application layer and processes them here. The SAP Web Application Server can run on:

- ▶ Oracle
- ▶ DB2
- ▶ Informix®
- ▶ MaxDB
- ▶ SQL Server®

Check the full matrix at:

<http://service.sap.com/pam>

When configuring an SAP system, you have to choose how to distribute the three layers among the available hardware based on the role of the system, expected workload, and available hardware. According to this distribution, the system has one of these three topologies, as described in the following sections.

One Tier

All the layers reside in the same server. Presentation, application, and database processes run on the same hardware. This configuration is typically used for development and test systems.

Two Tiers

This distribution is also used for development and test systems, but can be used for small production systems as well.

Here the application and database processes run on a server, and the presentation processes run on a different server.

Three Tiers

This is the right configuration for production systems. In this topology, all layers are distributed in different servers.

Several application servers can use the same database simultaneously, as well as several presentation servers using the same application server.

Multi Tiers

In this variation a fourth layer is added. The Web interface and external world is situated in this tier. A graphical view of the distributions is shown in Figure 2-2.

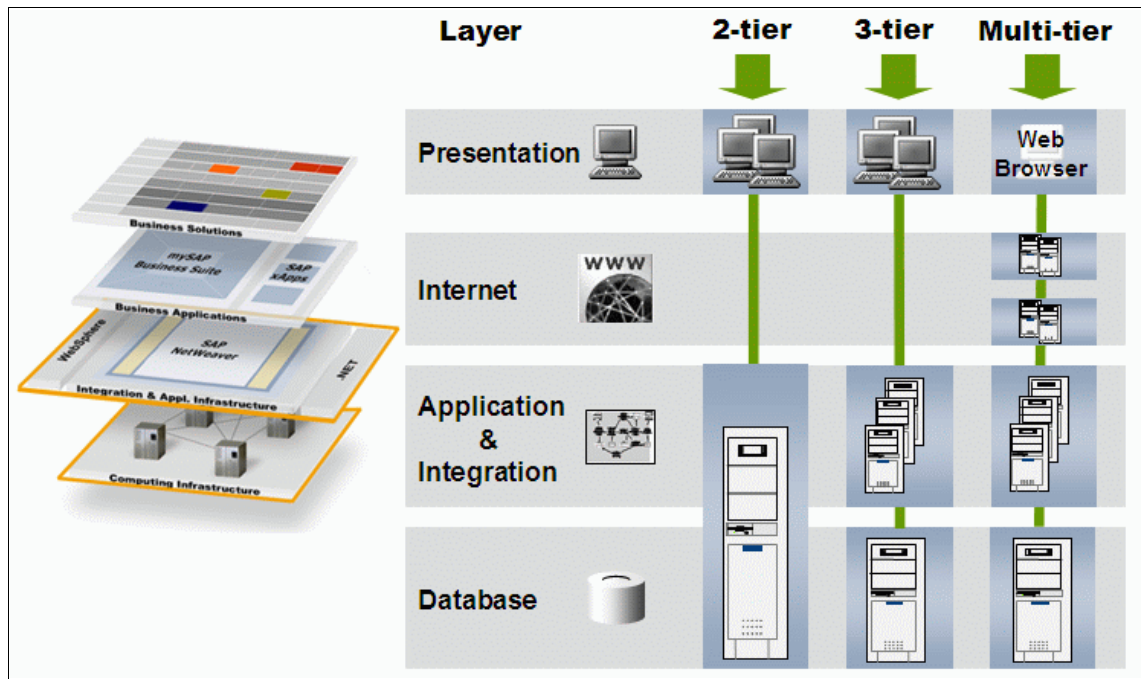


Figure 2-2 SAP NetWeaver layers

2.1.3 SAP NetWeaver and high availability

SAP NetWeaver high availability is about the elimination of single points of failure in an SAP system. There are some services in the SAP NetWeaver framework that cannot be replicated. They cannot exist more than once for the same SAP system, therefore, they are a single point of failure. For SAP NetWeaver 7.0, these services are:

- ▶ Central Services Instances (ABAP Central Services and JAVA Central Services)
- ▶ Database Instance
- ▶ Network File System (NFS)

With SAP NetWeaver version 7.0, there is a specific installation option for high availability. This installation option prepares the SAP NetWeaver to be installed in a switchover cluster.

A switchover cluster consists of:

- ▶ Redundant hardware to enable the ability to switch system resources from the primary hardware to the redundant (secondary) hardware
- ▶ A cluster management software to detect the failed resource(s)
- ▶ A method for achieving transparent switch for the SAP application

The installation of the high availability option is recommended by SAP in order to eliminate those single points of failure in the foregoing discussion.

2.2 SUSE Linux Enterprise Server

The target group of SUSE Linux Enterprise Server is the business community. There are several features in this distribution useful to the enterprise market, such as:

- ▶ Scalability up to 1024 processors
- ▶ Support up to 10 TB of RAM
- ▶ Support of the latest network technologies
- ▶ Virtualization
- ▶ High availability solution
- ▶ Graphical iSCSI management tools

2.2.1 Historical background

SUSE Linux distribution started in 1992 in Germany as a distribution based on Slackware. The former name was SuSE (Software- und System-Entwicklung).

During the development of the distribution, some well established tools of other distributions were incorporated into the project, such as the Red Hat Package Manager (RPM).

In 2003 Novell acquired the company that produced SuSE and since then has invested in the distribution, adding improvements and tools to increase its acceptance by the enterprise market.

2.2.2 SUSE Linux Enterprise Server and SAP

SUSE Linux Enterprise Server is designed to be an operating system for mission-critical systems such as SAP NetWeaver.

Since mid-2007, SAP recommends Novell SUSE Linux Enterprise Server as a preferable operating system between the Linux distributions.

SAP and Novell have an agreement to provide mutual support for SAP applications and the SUSE Linux Enterprise Server operating system. This enables customers running SAP on SUSE Linux Enterprise Server to receive support from the operating system to the SAP application from one central location.

More details about the available types of support can be consulted on:

<http://www.novell.com/products/server/sap.html>

That agreement includes a special Novell support package for the SAP Solution Manager. This support package enables SAP Solution Manager to also manage the operating system from a single point. Operating system updates should be applied recurrently and are available at the Novell customer center download area.

2.3 LifeKeeper for Linux

SteelEye's LifeKeeper for Linux is a software application that ensures the continuous availability of applications by maintaining system uptime. LifeKeeper maintains the high availability of clustered Linux systems by performing system and application health monitoring, maintaining client connectivity, and providing uninterrupted data access regardless of where clients reside; on the corporate Internet, intranet, or extranet.

To enable automatic system and application recovery if the system goes down, LifeKeeper allows applications to failover to other servers in the cluster. This helps LifeKeeper minimize the risk of a single point of failure and allows Linux systems to meet the stringent availability requirements of mission-critical operations by creating a fault resilient environment.

2.3.1 LifeKeeper for Linux components

LifeKeeper consists of three distinct components incorporated to ensure high availability:

- ▶ LifeKeeper for Linux Core
- ▶ Application Recovery Kit
- ▶ LifeKeeper GUI

LifeKeeper for Linux Core

The LifeKeeper core delivers the basic software infrastructure required to build a cluster. This includes packages to help recover a cluster database, cluster communications, and interfaces required by other LifeKeeper components.

The packages help maintain core components of the clustered system, such as the operating system, file systems, communication (IP) recovery, and interfaces required by other LifeKeeper components. The core bundle includes packages for the GUI administration and documentation. The core product also comes bundled with recovery software for core system components, such as the SCSI disk subsystem, and file systems and IP addresses.

Application Recovery Kits

Application Recovery Kits (ARKs) sit on top of the core and utilize application specific information required to perform health monitoring and recovery. There are Application Recovery Kits for Oracle, IBM DB2 UDB, MySQL™, SAPDB, Apache, Sendmail, and SAP, to name a few. While independent in the sense that each ARK contains specialized knowledge of its own application, they can be combined to build complex hierarchies with interdependencies.

In order to protect SAP systems, for example, the SAP Recovery Kit, which provides monitoring and switchover for the CI, is used in conjunction with the appropriate database (such as DB2), Application Recovery Kits for DB protection, and the NFS Server Recovery Kit for protection of the NFS mounts. The IP Application Recovery Kit would be used as well to provide for a virtual IP address that can be moved between NIC cards in the cluster as needed.

The various Application Recovery Kits are used to build what is called a “hierarchy” that provides protection for all of the components of the application environment. Each of them contain code that monitors the health of the application under protection and is able to stop and restart the application both locally and on another cluster server. The SAP Recovery Kit has been developed in consultation with SAP to ensure that it is using the most effective means for monitoring and recovering the SAP Central Services Instances and that all integration issues between the Central Services Instances, the DataBase, the Network File System, and the Application Server are accounted for in the recovery operations.

LifeKeeper GUI

The third component of the LifeKeeper architecture is the Graphical User Interface (GUI). The GUI is used to build the cluster, to define which applications/services are to be protected, to assign stand-by responsibility to appropriate nodes, and to monitor the cluster. Written in Java, the GUI can be

run on either the cluster systems themselves or from any browser that can access the cluster.

Together, these three pieces, the core, the GUI, and the associated Application Resource Kits, deliver a fully featured high availability product for Linux. LifeKeeper's customizable architecture makes it ideal for providing the protection required in SAP environments.

The three LifeKeeper for Linux components are shown in Figure 2-3.

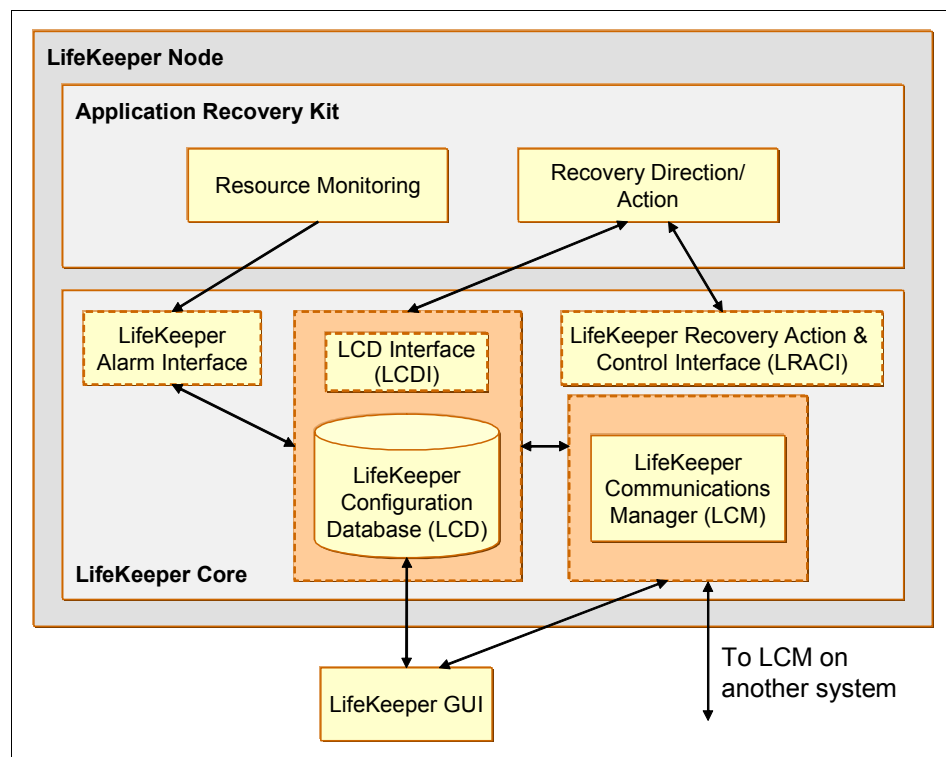


Figure 2-3 LifeKeeper for Linux components



High availability architectural considerations

A highly available architecture employs fault resilient hardware, automated failure detection, and recovery functions to ensure availability of business operations. From a technology perspective, the architecture is dependent on the business requirements and the products used to implement these business functions within the infrastructure. The solution architecture should be driven to maximize reliability and availability.

Architecting for high availability starts by first focusing on and preventing failures from occurring in the first place. Eliminating single points of failure with redundancy is a key characteristic in planning for high availability.

Building a solution for system wide availability involves designing each of the components for reliability. In this chapter, we cover the following components and highlight the various considerations for each in providing high availability:

- ▶ “Server”
- ▶ “Base software”
- ▶ “Network”
- ▶ “Storage”
- ▶ “Database”
- ▶ “SAP NetWeaver components”
- ▶ “SAP NetWeaver Single Points of Failure”
- ▶ “SAP NetWeaver in cluster configurations”

3.1 Server

Ensuring high availability of the server starts by implementing redundancy and providing reliable hardware. In a cluster, the first level of redundancy is to have an extra server. A second server provides the capability to take over resources even if the primary server fails.

In addition, redundancy should be designed to include the components within the server. This includes safe guarding communication (networks), power, memory and storage connectivity (multipathing).

This book features IBM System x and BladeCenters; however, SteelEye LifeKeeper and SAP NetWeaver can both exploit additional hardware platforms. For information on supported IBM hardware for Linux, refer to SAP Note 766222, which can be obtained from the SAP Marketplace by searching from this link:

<http://service.sap.com/notes>

Note: The SAP Marketplace is a secure site and requires a user ID and password.

Choosing the hardware platform for your SAP application depends on many factors and is not within the scope of this book. Capacity, workload characteristics, existing infrastructure, and availability requirements are just a few factors in hardware platform selection.

For assistance in sizing SAP on IBM Systems, contact the IBM Techline, ISV solution center. Refer to the following URL for detailed instructions:

<http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS2336>

The IBM System x and BladeCenter® servers are based on mainframe-inspired technologies to provide enhanced performance, availability, scalability, and power and cooling, while benefiting from the lower costs associated with industry-standard hardware and software. Based on IBM X-Architecture®, the System x servers are highly manageable and can help to reduce complexity in a timely and cost effective manner using either the Linux or Windows operating systems.

The BladeCenter has a single chassis that contains multiple hot-swappable blades, or independent servers, each with its own processors, memory, storage and applications.

This makes the blade server technology not only efficient in floor space but can help to simplify solutions by having a complete SAP ecosystem in one BladeCenter. Additionally it brings network and storage switching into a single enclosure, all tied together with powerful solution management tools.

The price and performance ratio, which is a key focus for SAP customers in the mid-market, is another perfect fit for IBM System x or BladeCenter.

Detailed information on IBM System x and BladeCenters is available in *The Benefits of Running SAP Solutions on IBM System x and BladeCenter*, REDP-4234, located at:

<http://www.redbooks.ibm.com/abstracts/redp4234.html>

Clustering allows two or more servers to provide a fail-safe environment through redundancy. This type of clustering can be referred to as a failover cluster, high availability cluster, or switchover cluster. The term *failover cluster* is often used in the context of database clusters. The term *switchover cluster* is commonly used in SAP NetWeaver environments and is synonymous with the term failover cluster in this book. The term *switchover cluster* describes the process of switching critical resources to an alternate server.

In this book, *high availability clusters* or *failover clusters* are terms used to reference both the database and physical servers themselves. A detailed definition of high availability clusters can be found at:

http://en.wikipedia.org/wiki/High-availability_cluster

Clustering technology can also provide scalability by load balancing and enabling the collection of servers to function as a single unit increasing the computing system capabilities. Clustering for scalability is often referred to as performance clusters or application clusters.

In an SAP NetWeaver architecture, the application server is an example of clustering for scalability. The application server can be configured with multiple instances (or occurrences) each on separate physical servers.

Clustering software can be used to reduce the impact of scheduled outages for firmware and software upgrades by moving the application and dependent resources off the system requiring maintenance.

The test environment used in this book consisted of two blade servers with identical configurations accessing shared storage. When a failure of a server occurs, the second blade server takes over the appropriate resources.

Having redundant servers provides resiliency at the server level. However, one single server consists of hardware components that might fail and might compromise their functionality either partially or entirely. To provide additional protection, servers should be equipped with high availability features for all major subsystems, like memory, disks, input/output (I/O) fans and power supplies. These components can be hot-swapped, so that a repair without cycling a server becomes possible.

3.1.1 Power supply

A failure of the power supply might be one of the most obvious problems. To prevent an outage caused by a failure on the power network, an uninterrupted power supply should be provided. This uninterruptable power supply should be sized according to the business requirements.

A single server can also be equipped with redundant power supply. This can resolve the dependency on one power network and prevent suffering failure from a single power supply by providing electricity through multiple paths. Verification should be performed to ensure that there are no shared components, such as fuses. Ground fault circuit interrupters are commonly overlooked.

The IBM Bladecenters provide dual power connections to each blade.

3.1.2 Memory

Failure of a memory module causes an application currently accessing the storage to crash. If that is a critical operating system process, there is a risk that the entire server might crash. To prevent such failures, IBM System x and BladeCenters provide many enhanced features such as:

- ▶ Chipkill™
- ▶ Memory ProteXion, also known as redundant bit steering
- ▶ Hot-swap and hot-add memory
- ▶ Memory scrubbing

In addition, if a memory module exceeds a threshold of errors, the data can be moved automatically to a new memory module.

Operating system independent memory mirroring can be used to hold two copies of all memory data similar to a RAID-1 disk mirroring. This allows hot-swapping of memory modules during a failure.

3.1.3 Ethernet network adapters

An Ethernet network adapter provides one unique connection to a local area network. A single network adapter is a single point of failure, as there is no availability of an application to the users through the network when it fails.

A server can be equipped with multiple network adapters. Today, many servers are delivered with at least two built-in Ethernet adapters. These can be configured to form a redundant pair. Often, these two adapters share one chip on the mainboard. So if one interface is damaged, it is possible that the second adapter is also impacted and does not work anymore. The addition of another network adapter as a plug-in card is recommended for high availability.

3.1.4 Storage Area Network adapters

Much like ethernet adapters provide connectivity to a Local Area Network (LAN), a high performance I/O channel provides server connectivity to a Storage Area Network (SAN). These devices are typically Fibre Channel, although SCSI is sometimes used. For high availability, these components can be built around hubs and switches to eliminate single points of failure. Given the network flexibility, redundancy can be configured in a variety of ways.

For high availability, multiple host bus adapters, multiple switches, and multiple fabrics should be implemented.

3.1.5 Central processing unit

A central processing unit or processor of a server is a single point of failure. Even if there are multiple processors, there are components such as I/O modules on the mainboard that cannot be redundant. Mainboards are known to be very reliable but remain a single point of failure.

3.2 Base software

To support the SAP NetWeaver environment, software at the infrastructure level is required — in particular, a reliable operating system, database software, and cluster management software to support high availability.

This section describes the base software stack used to support the underlying SAP NetWeaver architecture. In the following sections, we discuss the components of the SAP application.

The core software stack used to support the SAP environment included:

- ▶ Clustering Software
- ▶ Base Operating System Software

3.2.1 Cluster software

Cluster management software helps ease the complexity of a cluster environment by automating both the administration and monitoring. It provides a centralized console to quickly review and analyze resources. There are several flavors of cluster management software on the market today, including open source software.

SteelEye LifeKeeper is clustering software that maintains high availability by monitoring the health of critical system and application components, ensuring continuous availability. If a failure should occur, LifeKeeper automatically moves the protected resources to another server with reduced downtime. Automated cluster management is essential for high availability. The SteelEye LifeKeeper software is available for Linux and Windows operating systems.

LifeKeeper provides a cost effective solution to proactive protection of the software, applications, data and communication paths. LifeKeeper, combined with System x, can provide a low-cost alternative for customers looking for node failover on a platform that in itself provides a powerful solution at a relatively low price point.

Within an SAP environment, LifeKeeper can be used to protect several critical components. LifeKeeper can monitor and protect the database instance, SAP instances, network attached storage, and operating system I/O components. To ensure protection of all these components, a LifeKeeper installation includes a core package plus additional application kits.

These application recovery kits help in integrating resources and dependencies for the cluster. A set of resources can be grouped together, and the group can be managed by LifeKeeper as a single entity when starting or stopping.

Resources controlled by the cluster have to be enabled on at least one node. To achieve automatic failover, resources need to be enabled on a secondary node.

For example, a resource can represent:

- ▶ A cluster IP address
- ▶ A file system
- ▶ An application

Tip: An application is started on one node only. It is possible to install the application software on shared cluster storage or on local disks, and this ought to be an architectural decision. For example, you could decide to install the software binaries on local disks to help minimize downtimes for version upgrades.

Application Recovery Kits provide protection for additional components in the cluster. Storage related Application Recovery Kits facilitate the usage of:

- ▶ Disk mirroring
- ▶ Multi-path devices
- ▶ Individual SCSI disks
- ▶ Logical Volume Manager

A storage network and storage subsystem used by SteelEye LifeKeeper has to fulfill requirements for SCSI locking.

LifeKeeper uses SCSI2 reservations in single path environments or environments where multipath devices appear as single disk devices to the operating system:

- ▶ Single path drives called /dev/sdb, /dev/sdc, ...
- ▶ IBM RDAC
- ▶ Qlogic host bus adapter failover

LifeKeeper uses SCSI3 persistent group reservations in multipath environments, where the multipath layer introduces additional logical multipath devices:

- ▶ Linux Device-Mapper Multipath I/O
- ▶ IBM Subsystem Device Driver (SDD)
- ▶ EMC² PowerPath

Only certified storage subsystems can be used with SteelEye LifeKeeper.

A disk mirror consists of at least two disks and is represented by two resources. This is shown as a hierarchy in the administration console of the LifeKeeper software as depicted in Figure 3-1.

3.2.2 Base operating system software

An operating system is software that manages and enables resources on a server. A base operating system controls core tasks, such as allocating and releasing memory, prioritizing tasks, handling input and output devices, and managing files.

The Linux operating system has gained popularity over the last several years. There are several distributions of Linux software. IBM supports Linux, and the entire IBM Systems product line is Linux enabled. Figure 3-2 depicts the IBM server family supporting the Linux platform.

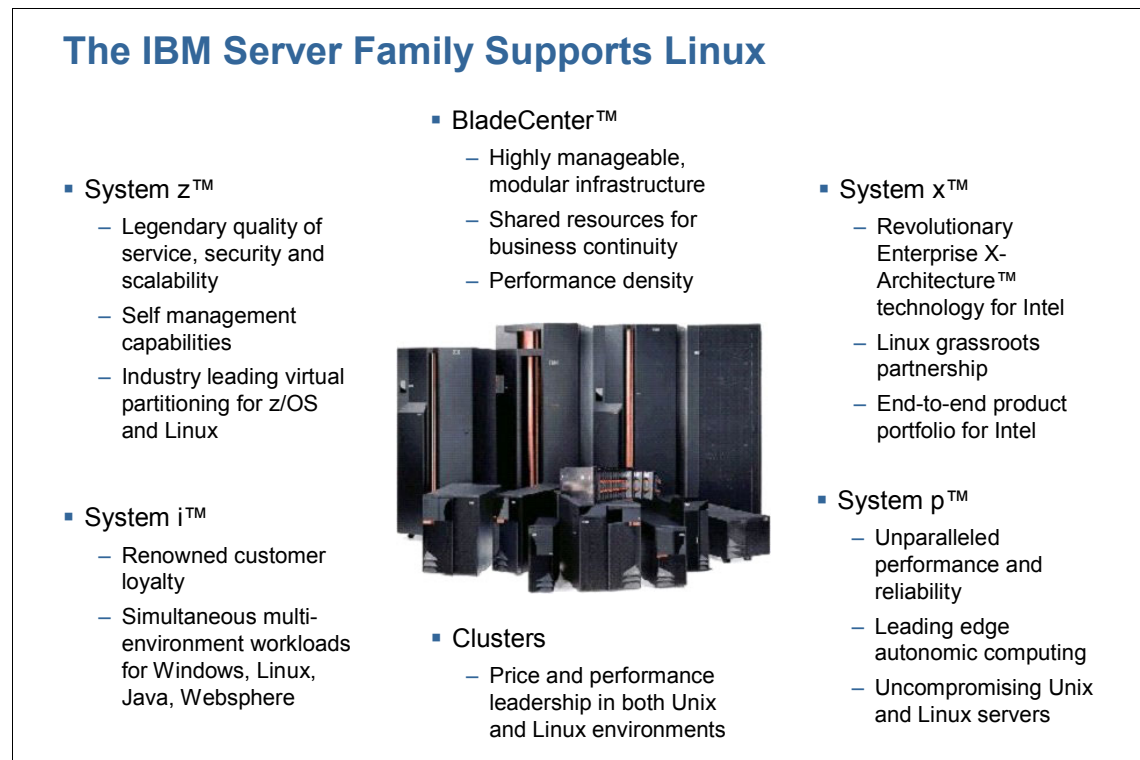


Figure 3-2 The Linux enabled IBM product family

The Novell SUSE Linux Enterprise Server provides an open, reliable platform that can easily scale as workload demands. Mission critical applications such as SAP are exploiting Linux technology to provide stable solutions. For a complete list of IBM hardware supported on the Novell SUSE Linux Enterprise Server, refer to the following Web site:

<http://developer.novell.com/yesssearch/Search.jsp>

Failover clusters are a common approach to extending availability for Linux. In the simplest form, a failover cluster has two nodes. The primary node is active and the secondary node stays on standby. In the event of a failure, the standby takes over resources to allow mission critical applications to continue. A failover cluster is an important requirement to ensure availability of the Linux application.

In terms of high availability, the redundant array of independent disks (RAID) is a technology that supports the integration of two or more devices for redundancy. RAID functionality can be provided from within a disk array or disk controller (hardware), or from the operating system (software).

SUSE Linux Enterprise Server offers the ability to combine several physical disks into one virtual device with Software RAID. Software RAID is an inexpensive solution offered by many operating systems. It shifts the disk management functions off the hardware, allowing for a more flexible choice in hardware selections. Software RAID functions are performed on the host, and thus, there is an overhead incurred on the system.

The testing scenarios in this book leveraged both failover clustering and Software RAID for Linux. This approach provided a cost advantage and increased flexibility by removing hardware limitations.

3.3 Network

The network connectivity of a high availability cluster requires special considerations both for the cluster internal communication (heartbeat) and client accessibility to an application running in the cluster. It can always happen that data packets get lost on the network or that components on the network fail or require maintenance. Client connections must be safeguarded by providing some level of transparency to network addresses.

Having two network cards connected to the same network switch for a failover configuration, eliminates the network card and the port as an SPOF. However, a failure of the physical switch or the uplink connection results in a failure.

To increase network availability, a connection to a separate switch can be used. This switch can have a direct connection and a connection to the uplink. Therefore, if either fails, the server can be reached by the other. Additional network setup is required using a special mechanism that prevents loops.

Attention: The OSI layer-2 Spanning Tree Protocol (STP) uses an algorithm to provide path redundancy while protecting against link loops. Without STP, it is possible to have two connections that can be simultaneously live and can result in an endless loop of traffic on the network. However, STP is not recommended for network ports that are used for the cluster heartbeat communication. STP configuration settings can cause the network ports to close when computing a new tree configuration after failure. Although it is possible to increase the heartbeat default, this will increase the reaction time for failures. For STP to be compatible with SteelEye LifeKeeper, the time-out and retry settings for the IP Recovery Kit will need to be adjusted (/etc/default/LifeKeeper).

Figure 3-3 shows an example of a redundant network connection for two servers located in different server rooms. According to the Open Systems Interconnection Basic Reference Model, the second layer of a Local Area Network connection is the data link. This link can be either for an application's client communication or for a heartbeat. There should always be dedicated network cards for a heartbeat.

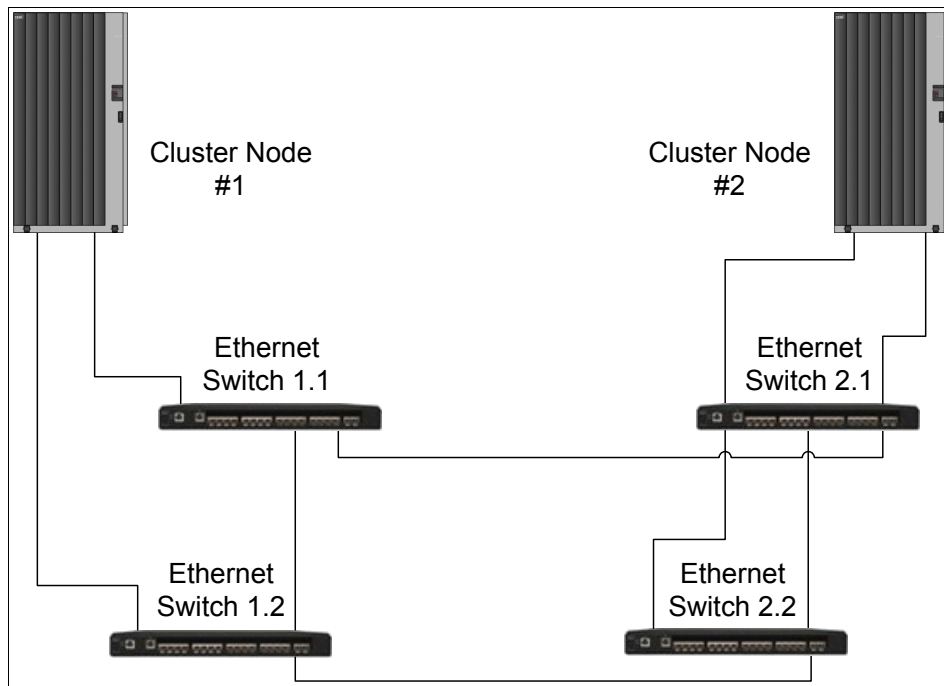


Figure 3-3 Redundant network connection

Figure 3-4 shows the seven layers of connections developed in the Open Systems Interconnection (OSI) reference model.

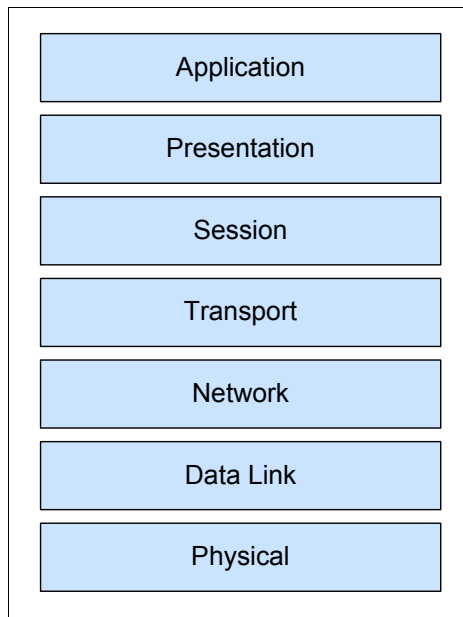


Figure 3-4 Open Systems Interconnection (OSI) Basic Reference Model

While it is possible to achieve a failover on that data link layer, it is commonly implemented on the network layer. Today, TCP/IP is standard on the network layer. At the data link layer, standard ethernet (Fast or Gigabit Ethernet) according IEEE 802 should be considered.

A device on the data link layer, such as a network plug-in adapter, has a Media Access Control address (MAC address). There can be multiple TCP/IP addresses behind a MAC address. The Address Resolution Protocol (ARP) provides MAC addresses for TCP/IP addresses. ARP uses broadcast packets and can be used between devices on the same data link layer only.

One to many network switches connect multiple devices on the same data link layer. A switch needs to know on which physical port a MAC address can be found. If a network port on the switch fails, the cable can be plugged in to an alternative port on the same switch. To achieve this automatically, a server could be equipped with two network cards (NIC), both connected to a switch, and the server can move the Mac address from one to the other.

An alternative to this solution, is to use takeover of the TCP/IP address, leaving the MAC address untouched.

3.3.1 MAC address failover

If the Media Access Control address of an interface can be set through a driver, the address of the failing adapter can be carried to a secondary adapter. By sending out an Ethernet packet through the secondary adapter, the network switch is informed that the position of the interface card has changed. This is very much like unplugging a network card and plugging it into a different port. The impact of moving the address is almost transparent to other devices on the network.

Finally, two or even more network adapters can be used as a trunk. This allows a higher overall bandwidth from a server into the network, but requires an additional configuration on the network switches. At the same time, it gives the fastest reaction on a failure of a single adapter.

Uniqueness is required for both Ethernet media access control addresses and IP addresses. Otherwise, conflicts occur.

A takeover of a MAC address makes the originating adapter useless. While it is possible to have multiple IP addresses behind a MAC address, it is not advantageous. MAC address failover offers no advantages compared to IP address failover when an application restart is required.

3.3.2 IP address failover

It is possible to have multiple TCP/IP addresses on the same Ethernet adapter. These could be a native machine address and a specific alias for a particular network service. This service address can be taken down if necessary and brought up on another network card or even to another machine.

When moving an IP address to another interface with a different Ethernet hardware address (Media Access Control address), it can take a few minutes until all affected devices on the same network recognize the change.

Uniqueness is required for both Ethernet media access control addresses and IP addresses. Otherwise, conflicts occur. This could be accelerated by sending out a broadcast packet that informs all connected devices and is called gratuitous ARP.

A takeover of one IP address to another network card or another machine allows all other addresses on the same network card to remain available. Under normal circumstances, moving an IP address from one cluster node to another in a failover configuration causes no significant delays.

3.3.3 Redundant network switch architecture

An Ethernet switch keeps a table containing all MAC addresses for a network port. If two network switches are connected directly, they recognize which MAC addresses can be found on the opposite switch through broadcast messages. If a third switch is added and connected to the first two, a loop is created. That condition can cause the network to become almost unusable.

The spanning tree protocol was designed to recognize switches or bridges on the data link layer, and to allow automatic recognition of the shortest path for communication between the devices on one network.

The Spanning tree protocol allows a network topology in Figure 3-3 where all switches are connected to each other, one server is connected to one switch with one network interface, and if either one switch or network card fails, all others are still able to communicate.

We recommend having an independent communication path that is not part of a Spanning Tree and working as independently as possible. This could be a dedicated cluster heartbeat network or a virtual local area network across switches without any dynamic configuration.

3.4 Storage

Data is one of the most valuable assets a business has. The inability to access data or the loss of data translates into revenue loss. An SAP solution is no exception to that rule; data is essential to the application. SAP landscapes rely heavily on system data stored in file systems and application data stored in relational databases.

That makes a storage system a strategic investment of many businesses. There are several storage technologies available as a solution:

- ▶ Direct Attached Storage (DAS)
- ▶ Network Attached Storage (NAS)
- ▶ Storage Area Network (SAN).

NAS and SAN have really captured the market as customers strive to exceed the limitations of Direct Attached Storage (DAS). There have been many articles and white papers published discussing the comparison of these technologies. This book does not cover the comparison or selection of the storage technology, but rather, highlights some of the features that can be used for HA architecture.

A highly available system requires a highly available storage infrastructure. To achieve high availability, data must be accessible from different nodes (servers). It might also be necessary to store redundant copies of this data. Redundant copies might have to be stored both within a local site, and at an offsite or geographically disperse site for disaster recovery requirements.

To achieve data consistency between these redundant copies, replication technologies can be used.

3.4.1 Direct Attached Storage (DAS)

Direct attached storage is managed individually by the servers. It is a term used to describe a storage device that is directly attached to the host. In its simplest form, it is the internal disk of a server. DAS technology has limitations when the requirement to scale arises or if it is necessary to move data around the network. DAS storage also offers the least availability of the storage technologies. A failure at a server typically means that the data attached to the server is unavailable.

3.4.2 Network Attached Storage (NAS)

Network attached storage solutions make use of a local area network through which a server can connect. Since the storage is independent from the server, it can provide different subsets of data to multiple clients through different protocols.

For example, a part of the disk can be shared to Microsoft Windows based clients via the Microsoft Common Internet File System (CIFS) protocol, while another is used from a Linux machine using the Network File System (NFS). Both CIFS and NFS allow, in general, concurrent usage of files.

NAS storage enables centralized management, backup, and security. NAS brings a level of fault tolerance to storage by allowing multiple servers accessibility. In contrast to a DAS environment, when a server fails, the data that server holds can be made available through a secondary server.

Today, the Internet Small Computers System Interface (iSCSI) is becoming more popular as implementations became part of Linux and Microsoft Windows operating systems. Via the iSCSI protocol, a volume is accessed like a SCSI disk but over a local area network. As an iSCSI volume is a block device, replication mechanisms similar to physical disk replication (like RAID) can be used to avoid having the network attached storage as a single point of failure.

3.4.3 Storage Area Network (SAN)

A SAN is a dedicated high performance storage network that transfers data between servers and storage devices, separate from the local area network. Unlike NAS, which utilizes TCP/IP, SANs often utilize Fibre Channel. SAN based storage enables data to be backed up directly to SAN attached tape subsystems so LAN resources are not required.

SANs enable storage to be independent of applications and accessible through multiple data paths for better reliability and availability. Centralized and consolidated storage also allows for simpler management. DAS or NAS are typically optimized for sharing at file system levels; a SAN has the ability to move large blocks of data. A SAN provides fast data transfer while reducing I/O latency and server workload, because it offloads the work from the network and can provide “LAN free” operations.

When SAN based storage was first introduced, it was mainly used to store application data. However, SAN based storage today can be used to protect much of the data traditionally stored on internal drives. The IBM System p and x family introduced the ability to boot from the SAN in 2002. This simplifies management of a data center, and allows administrators to quickly replace failed servers without reconfiguring arrays or restoring data.

A SAN uses special switches as mechanisms to connect the storage devices. Since a SAN requires new hardware, it introduces a level of complexity in maintaining the infrastructure. For high availability, these new hardware components must be implemented with redundancy in mind. Multiple host bus adapters, multiple switches, and multiple fabrics are each required to eliminate single points of failure within the SAN topology.

A SAN usually contains the following components:

- ▶ Server Fibre Channel host adapter (HBA)
- ▶ Fibre Channel switches (FC switch)
- ▶ Storage subsystems
- ▶ Multi-pathing software on the host servers accessing the storage

Figure 3-5 depicts SAN components to be considered in designing high availability.

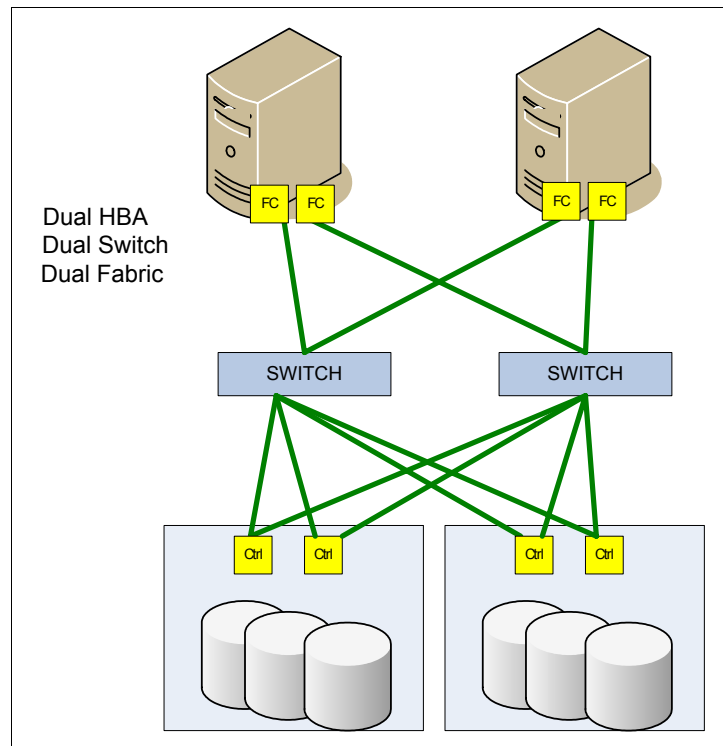


Figure 3-5 SAN Components with two arrays

This diagram shows a storage subsystem that consists of many disks and provides segments of that space in various ways through clients on a storage network. These disks can be grouped into arrays following the RAID standard, depending on requirements. Currently, the following levels are fairly common:

- ▶ RAID 0 (striping) concatenates several disks to one large volume without any redundancy. The controller splits the I/O among the disks, creating parallel write or read operations for performance improvements without protection.
- ▶ RAID 1 (mirroring) provides a simple synchronous copy of data on two disks. The controller performs the write operations in parallel. Read operations are improved because the controller uses all devices reducing contention.
- ▶ RAID 5 (striping with checksum) provides a mechanism of protecting data while striping, allowing volumes bigger than one disk and protecting data against failure of a single disk. It provides redundancy at the expense of one physical disk per array.

- ▶ RAID 10 (or RAID 1+0) uses a combination of striping and mirroring. Individual disks are mirrored and then these grouped mirrors are striped. Although RAID 10 is expensive, it provides a performance and fault tolerance sometimes required for database applications.

While RAID 1 provides a high level of protection and performance, it consumes half of the space of the disks. RAID 5 stripes the data across disks and uses checksums, allowing one disk in a group to fail without losing data. However, there is still a write penalty for RAID 5, and therefore it is sometimes not chosen for OLTP databases with random write intensity.

A SAN may also include SAN volume controllers (SVC) if storage virtualization is used. SVC controllers should be deployed in node pairs, with each node pair connected to different uninterruptable power supplies (UPS) to ensure availability. If a SAN is not implemented, FC switches are not in the I/O path but the principles of HA remain the same.

3.4.4 Multipath Storage Area Network connection

Similar to a Local Area Network, multiple host connections into a Storage Area Network can be achieved with multiple host bus adapters in a server. In contrast to the Local Area Network, communication paths are controlled by a driver layer on the host using the storage volume.

Figure 3-6 shows an example connection with multiple paths to the same storage subsystem. The volume is a LUN located in the DS8000™ storage subsystem. This LUN is visible from one host bus adapter to two switches and to storage subsystem host bus adapters, making it appear eight times in the operating system.

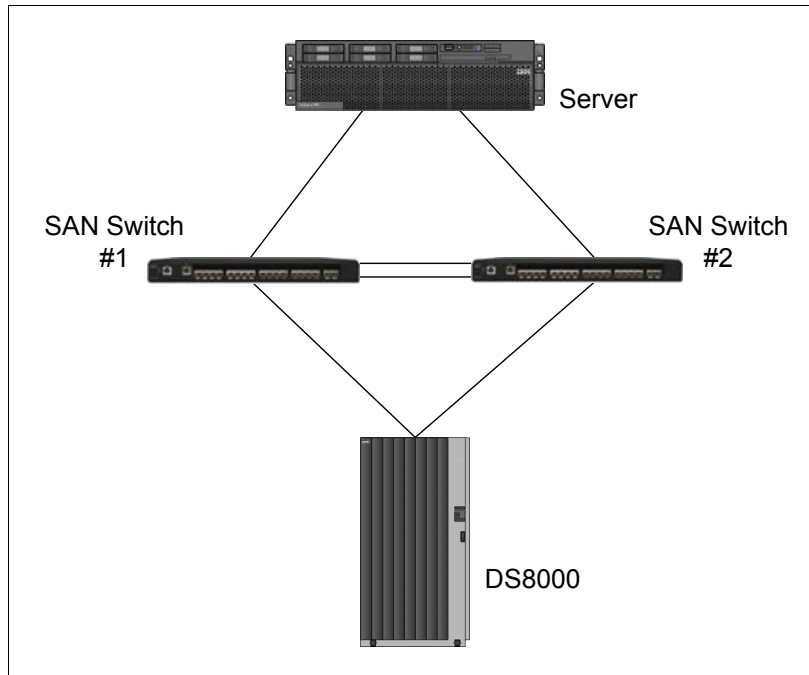


Figure 3-6 Redundant Storage Area Network connectivity

Modern storage subsystems have built-in high availability features within the components. These include:

- ▶ Hot swappable dual storage controllers
- ▶ Dual power
- ▶ Redundant fans
- ▶ Multiple disk and host adapters
- ▶ Multiple drives using RAID technology to increase availability
- ▶ Hot swappable spare drives
- ▶ Battery backed cache

Even with many built-in features, the storage subsystem itself is a SPOF. Like servers, this SPOF can be eliminated with redundancy by implementing a second storage subsystem.

SAN based storage subsystems can replicate (mirror) data, synchronous or asynchronous, between each other to provide redundant copies. As previously noted, IBM provides either Metro mirroring or Global mirroring technologies.

To provide high availability, a common technique is to duplicate storage with mirroring or replication. The testing environment for this book did not include a secondary storage subsystem, so the storage subsystem was an SPOF. In a critical SAP environment or when disaster recovery is a requirement, consider using storage based mirroring technology.

3.4.5 Replication

There are several solutions on the market today to replicate disk across a network or subsystem. Replication can be host based, network based, or array based.

Host and network based replication

For Linux operating systems, host based and network based replication can be achieved using any of these technologies:

- ▶ Distributed Replicated Block Device (DRBD)
- ▶ SteelEye LifeKeeper Data Replication
- ▶ Software RAID (md)

Both DRBD and LifeKeeper provide mechanisms to replicate data through a network interface to another node, either synchronous or asynchronous. Usually only one copy of the data is active at a time and can be used for write purposes.

SteelEye LifeKeeper Data Replication integrates well with the SteelEye LifeKeeper cluster manager. It is also available for Microsoft Windows operating systems, as well as Linux operating systems.

Software RAID implements the various RAID levels in the kernel disk (block device). It is sometimes referred to as operating system based RAID. A software layer provides an abstraction layer between the logical and physical drives.

These host based replication techniques use processor resources on the host server attached to the storage. These techniques are simple and easy to use. The DRBD and LifeKeeper technologies are synchronous over the network, so large amounts of data increase network traffic. Because SAP databases usually contain large amounts of data, our testing scenarios did not feature LifeKeeper replication.

Array based replication

Array based replication offers a performance advantage in replicating large amounts of data. Array based replication techniques are independent of the operating system and therefore do not consume resources on the host.

Array based replication creates continuously updated remote copies of the disk subsystem at either local site (campus), metropolitan, or geographically disperse sites. It supports large volumes of data and is mirrored at the disk subsystem microcode.

IBM offers two array based replication (mirroring) solutions:

- ▶ Metro Mirroring
- ▶ Global Mirroring

Metro mirroring is synchronous operation. The I/O operation to the host server ends after the second copy of the data is written to the replicated volume.

Global copy is asynchronous, so greater distances can be achieved. The I/O operation is finished when the data is written to the primary device and the data is asynchronously moved to the replicated device. Thus, there is a delay of typically 3-5 seconds in the replicated devices.

Figure 3-7 is a graphical depiction of the I/O timing in Metro and Global Mirroring techniques.

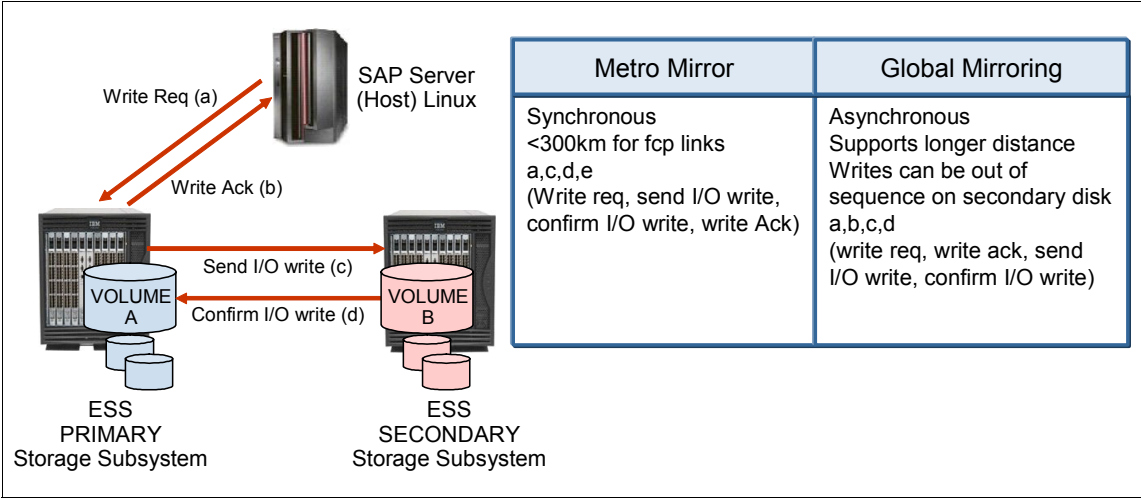


Figure 3-7 I/O timing in Metro and Global Mirroring

FlashCopy is an additional IBM feature that is available for disk storage subsystems. It enables the ability to make almost instantaneous point-in-time copies of entire logical volumes. For an SAP system, FlashCopy can be used to make backups and reduce the impact of traditional backups on the production server. FlashCopy can also be used to aid in cloning SAP environments. Cloning SAP environments is often a necessary and frequent requirement.

More information on the IBM storage product line and available copy services is available at:

- ▶ *IBM System Storage DS8000: Copy Services in Open Environments*, SG24-6788
<http://www.redbooks.ibm.com/abstracts/sg246788.html>
- ▶ *IBM System Storage DS6000 Series: Architecture and Implementation*, SG24-6781
<http://www.redbooks.ibm.com/abstracts/sg246781.html>
- ▶ *IBM System Storage DS4000 and Storage Manager V10.10*, SG24-7010
<http://www.redbooks.ibm.com/abstracts/sg247010.html>

3.5 Database

SteelEye LifeKeeper supports multiple RDBMS vendors, including Oracle, SAP MaxDB, Microsoft SQL Server, Informix and DB2 for an SAP NetWeaver environment. IBM's DB2 database for Linux, UNIX, and Windows (LUW) was featured in this book.

To design and implement a highly available solution for DB2 LUW, several methods are available. DB2 LUW can be configured to provide failover either through a high availability cluster, leveraging DB2 High Availability Disaster Recovery (HADR), by maintaining a standby database built with log shipment, or by maintaining a standby database with a replication techniques.

3.5.1 Replication techniques

Database replication involves two copies, a primary and a secondary (standby). It can be achieved by using either software based techniques or storage subsystem replication. Software techniques for replication, such as DB2 SQL replication or Websphere Q replication propagate committed transactions from a source to a target with a capture and apply technique.

DB2 HADR is a replication solution in which two separate DB2 databases, each with their own storage, are defined and synchronized. HADR ships logged transactions between the two databases with TCP/IP. HADR can provide faster failover times than clustered solutions, because HADR does not require disk takeover time. Replication technologies, such as HADR, offer an additional advantage in allowing rolling fixpack upgrades. Automatic Client Re-route (ACR) can also be leveraged to re-route connections with minimal outage.

For detailed information on configuring SAP with DB2 HADR, refer to the following article on developerWorks available at:

<http://www.ibm.com/developerworks/db2/library/techarticle/dm-0508zeng/>

The HADR technology has been incorporated into Informix Data Server (IDS) since 1990. Both SAP MaxDB and Oracle provide similar replication or standby technologies available on Linux platforms.

The IBM Global mirroring or Metro mirroring features can be used for disk based replication and to protect both the database and file systems. Global mirroring is a micro-code feature that supports asynchronous replication. Metro mirroring is a synchronous implementation.

The advantage of Metro mirroring is that there is minimal host impact for performing the replication, as the process is performed by disk hardware. The disadvantage is that there are possible host delays because the distance increases since it is synchronous. The greater the distance between the primary and secondary storage subsystems, the longer it takes each I/O to complete. So, for long distance, asynchronous replication would be a better fit. This book covers replication topics for SANs in more detail in 3.4, “Storage” on page 32.

3.5.2 Cluster techniques

In contrast to replication methodologies, in a typical failover (HA) cluster, there is a single copy of data (database) that resides on shared storage. A cluster IP address is used for client access. Failover clusters are fairly common solutions for local HA availability. In a failover cluster, the secondary server needs to acquire resources for the shared storage, start the DB2 instance on the failover (standby) server, and perform crash recovery. The recovery time depends on the time for the cluster software to detect the failure, the storage, and IP address take-over time, and the size of the logs required for crash recovery. This can typically take several minutes.

Figure 3-8 shows a graphical representation of clustering versus replication.

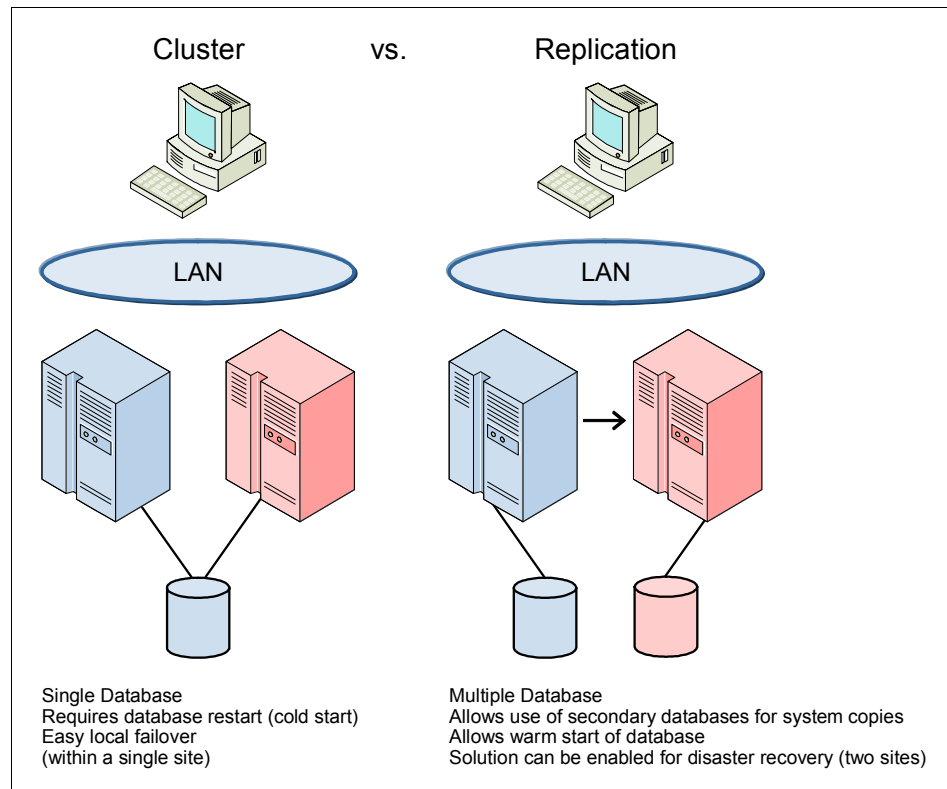


Figure 3-8 Clustering databases compared to replication

Clustering software can help provide sophisticated features to manage and control the resources. In this book, LifeKeeper is the operating system independent software used for cluster management. At the time of writing this book, LifeKeeper was not certified using DB2 HADR, so a traditional failover architecture with shared storage was implemented.

In designing a highly available solution for your mission critical database, any one or a combination (hybrid) solution can be used, depending on your particular requirements. Clustering is often chosen to provide high availability within a local site or single site failover. This is commonly used in conjunction with a replication technique to extend high availability to additional sites and to provide a redundant copy.

Figure 3-9 represents a commonly used hybrid solution.

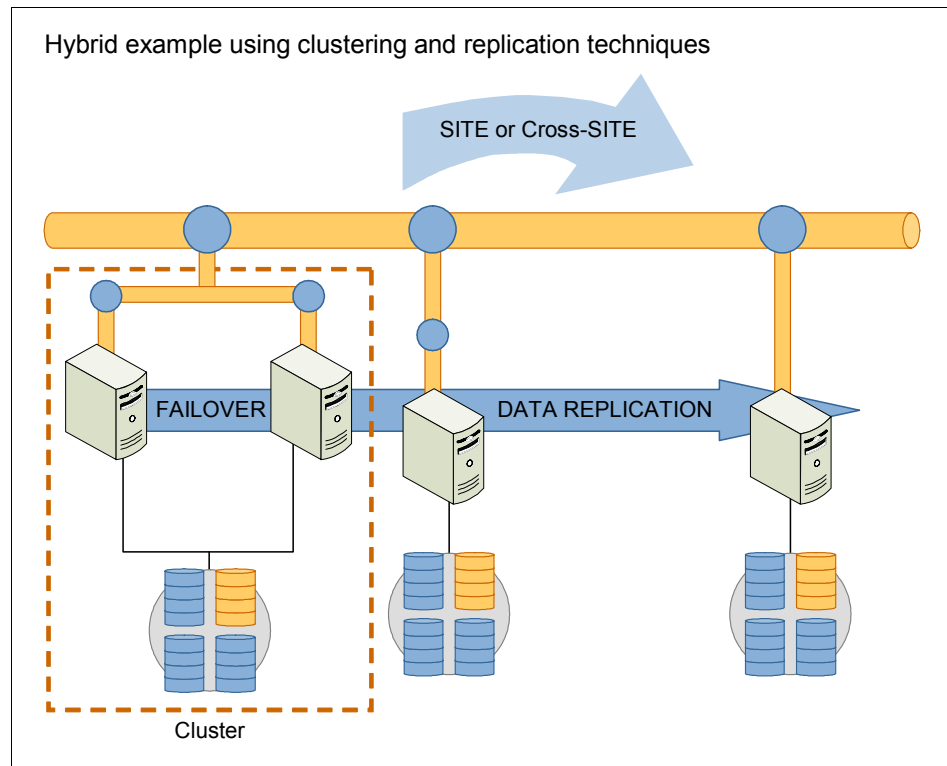


Figure 3-9 Hybrid HA solution

For more detail on high availability techniques for DB2, refer to *DB2 9.1 Data Recovery and High Availability Guide and Reference*, SC10-4228. This publication can be found at the DB2 Information Center or at the following URL:

<http://www-1.ibm.com/support/docview.wss?uid=pub1sc10422800>

In choosing the appropriate HA methodology for DB2, you have to take several factors into account.

This book focuses on building a cost effective architecture using LifeKeeper for Linux to ensure database availability. The architecture uses database failover (HA) clustering with LifeKeeper as the software that monitors and maintains database availability. LifeKeeper also helps to manage the IP addresses for client connections. The process of selecting an HA technology for a database solution was not included in the scope of this book. However, concepts have been included that can be implemented in other solution designs.

In a failover or HA cluster, redundant servers are configured to provide service when a component fails. DB2 LUW uses a shared-nothing architecture rather than the shared data available on DB2 for IBM System z. This means that at failure, a loss of access to all data managed by that server occurs until those resources can be failed over to an alternate server or restarted on the current server. Figure 3-10 depicts a typical two-node database failover cluster.

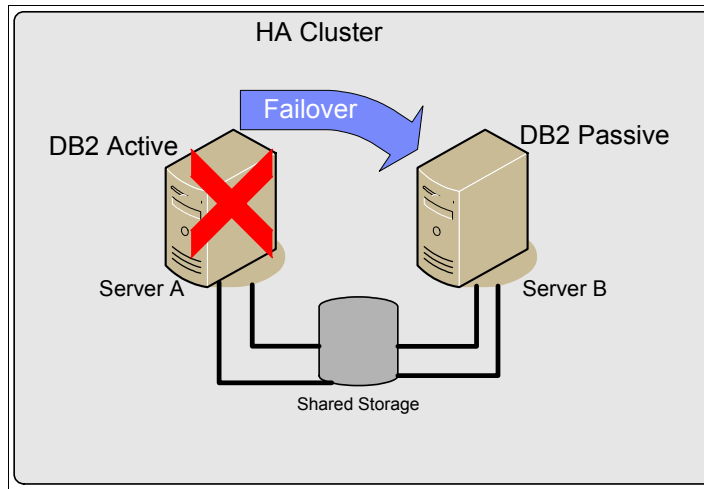


Figure 3-10 Failover or HA database cluster

LifeKeeper protects the database by monitoring the health and initiating automatic instance restart when a failure is encountered. LifeKeeper attempts to restart the instance on the primary server first. In the event that the database cannot be restarted on the primary (original) server, then LifeKeeper switches the resources to the failover server. This fault resiliency is completely automated through the LifeKeeper database (DB2) recovery kit.

At instance restart on either the failed or alternate server, crash recovery must be initiated. Crash recovery is the process of applying logs to bring the database back to a consistent state after the failure. DB2 has a configurable parameter, `AUTORESTART`, which can be used to automate crash recovery. When `AUTORESTART` is enabled, the database automatically activates on the first application connection after instance startup, applying crash recovery for any inflight transactions. This feature is often used in clustered (non-HADR) environments.

3.5.3 Additional considerations

LifeKeeper can optionally be configured to protect the DB2 administration server (DAS). The administrative server is not typically considered a requirement for high availability, since administration can be performed at the command line if the server is not available. In this book, configuration of DAS for high availability was not implemented.

All of the SAP application data is stored in the database. So, availability of the database is a critical component. Providing redundancy through failover is important for a HA architecture.

More information on the high availability for DB2 LUW can be found in the *High Availability and Scalability Guide for DB2 on Linux, UNIX, and Windows*, SG24-7363, using the following link:

<http://www.redbooks.ibm.com/abstracts/sg247363.html>

Keeping current with SAP certified maintenance levels can also alleviate known problems. As with SAP software, each maintenance release of DB2 provides additional fixes and enhancements based upon prior customer problems and requirements. This helps minimize exposure to known problems. Refer to SAP Note 101809 for up-to-date information on supported DB2 versions and fix pack levels, which can be obtained from the SAP Marketplace by searching from this link:

<http://service.sap.com/notes>

Note: The SAP Marketplace is a secure site and requires a user ID and password.

3.6 SAP NetWeaver components

SAP NetWeaver is a technology platform that provides an abstract layer between the underlying infrastructure components and the SAP business solution. It provides the capability to have many different SAP products and applications work together. This enables an SAP environment to run across different hardware, operating systems, and multiple relational database systems.

Within an SAP architecture, several different SAP instances may exist. In addition to the data instance, each of these SAP infrastructure components and instances need to be considered in the design for high availability. An SAP instance is a group of processes that are started and stopped at the same time.

With SAP NetWeaver 7.0, the message server and the enqueue work process were moved to its own instance. These services are called the ABAP System Central Services (ASCS) and System Central Services (SCS), respectively. The benefit of having these separate is mainly for availability. All instances within SAP are identified by a number and can reside on several servers.

Within a HA SAP architecture, the following instance types and infrastructure components exist:

- ▶ Central Instance (CI)
- ▶ Central services for Java (SCS) and ABAP (ASCS)
- ▶ SAP Central file systems
- ▶ Network File System (NFS)
- ▶ Application Server (AS)

3.6.1 Central instance

Every SAP system includes at least one central instance. The central instance contains numerous software components including, a database gateway, dispatcher, work processes for dialog, batch, spool or update, Internet communications, Internet graphic service, and software deployment.

3.6.2 Central services

The central instance services handles communication and synchronization for the Java and ABAP clusters. The central services consist of message and enqueue servers. The message server handles communication between the dialog instance and supplies data to the SAP Web dispatcher about load balancing. The enqueue server controls locking and synchronizes data in a Java cluster.

Locking is a key element for integrity; it prevents concurrent users from accessing inconsistent data. For example, by getting a lock for a database record before modifying it, you prevent any other user transaction from modifying the same record at the same time.

Locking in SAP is performed by the enqueue server. It is the responsibility of the application to request an enqueue for an object before accessing it. Updates to the database made on behalf of an application are not visible to other applications until the updates are committed, either implicitly or explicitly. The application requests this type of lock isolation to prevent the occurrence of dirty reads. If the application terminates abnormally, then all changes to the database made since the last commit point are rolled back.

The processing between the start of the transaction and the commit point is called a logical unit of work (LUW). Database integrity is maintained by ensuring that all changes to a database made during a LUW are either committed or rolled back.

To balance workload, it may also be necessary to distribute the database instance workload from ASCS onto two different servers.

3.6.3 SAP Central file systems

Several file systems are shared among all the instances in SAP. The filesystems hold SAP user and system data. There is also instance specific data stored on the /usr filesystem. SAP requires shared access to some directories, while for other directories it is optional; for example, the directories containing the executables.

Every SAP instance requires file systems to store temporary files, instance logs, and other system data. These data are stored in shared and local file systems.

In the shared file system, SAP stores common information used by all instances of a single SAP system, the SAP profiles, SAP job logs, SAP kernel libraries, and executable files. On UNIX and UNIX-like platforms, this file system is exported from the Central Instance server under the path, /sapmnt/<SID>.

At the local file system, SAP stores several logs and traces and instance specific files. This file system is mounted in the mount point, /usr/sap/<SID>”.

3.6.4 Network File System

To meet the requirements of shared directories in Linux, this information is shared on a Network File System (NFS). SAP information such as profiles, joblog, repository, and request data for transports is shared on NFS. This information is required by the central instance in SAP and is therefore critical for availability.

3.6.5 Application server

The application server (also referred to as the dialog instance) is optional and can be installed on separate servers. Multiple application servers are typically implemented for scalability, and it is common to have them reside on separate physical servers. Like the central instance, the application server instance handles numerous components including dispatcher, work processes, gateway, Internet communications and Internet graphic services.

Multiple application servers do not guarantee session persistence. SAP does not recommend session persistence or failover for application servers. However, it can be implemented. For more information on designing session persistence and AS failover, refer to the J2EE Application Management section of the SAP NetWeaver 2004 product documentation. The SAP NetWeaver 2004s product documentation can be found at:

<http://service.sap.com/nw2004s>

3.7 SAP NetWeaver Single Points of Failure

Similar to hardware redundancy in the previous sections, it is important to identify and eliminate any SPOFs for critical SAP processes. The four critical components identified as a SPOF in an SAP environment are:

- ▶ Central services for Java (SCS)
- ▶ Central services for ABAP (ASCS)
- ▶ SAP Central File System
- ▶ Database

Figure 3-11 is a graphical representation of these system-wide points of failure.

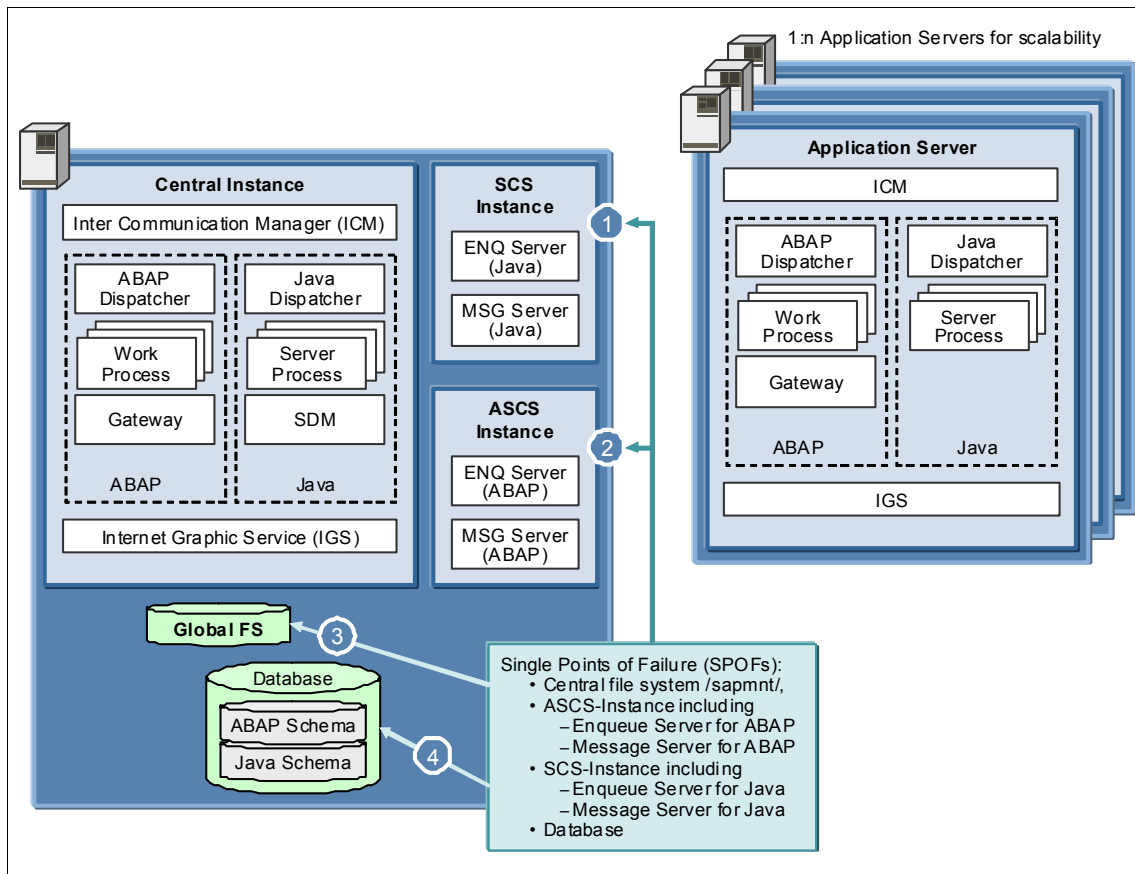


Figure 3-11 SPOFs within the SAP components

With the introduction of SCS, the term central instance has become obsolete. The software deployment manager (SDM) is installed on the central instance. However, the SDM is not typically considered a critical component for production systems, because deployment usually implies downtime.

3.7.1 Failure of the enqueue server

The enqueue server of the central services instance is a critical component. The enqueue server maintains database consistency by managing transaction locks. If a failure of the enqueue server occurs, transaction locks that are not committed must be released and uncommitted units of work are rolled back.

For resiliency, it is recommended that an enqueue replication server be implemented. A replicated enqueue server protects the system from loss of the lock table, ensuring a complete unit of work.

The replicated server should reside on a different server. If you have multiple Application Servers (AS), you can place the replication server on either one of the AS servers or the failover (standby) server. The replicated enqueue server contains an exact copy of the lock table.

If the standalone enqueue server fails, it is started by the LifeKeeper on the host machine in which the replication server is running. The replication table stored on the replication server is transferred to the enqueue server and the new lock table is created from it.

If the replication server fails, it can be restarted on a different machine. It retrieves the replication table from the standalone enqueue server when it restarts. When the replication server is running, it is supplied only with delta information arising from each request to the enqueue server.

There are typically two servers implemented for high availability of the enqueue service — one for the stand alone enqueue server and the second for the replicated enqueue service. Figure 3-12 is a graphical representation of high availability for the enqueue server.

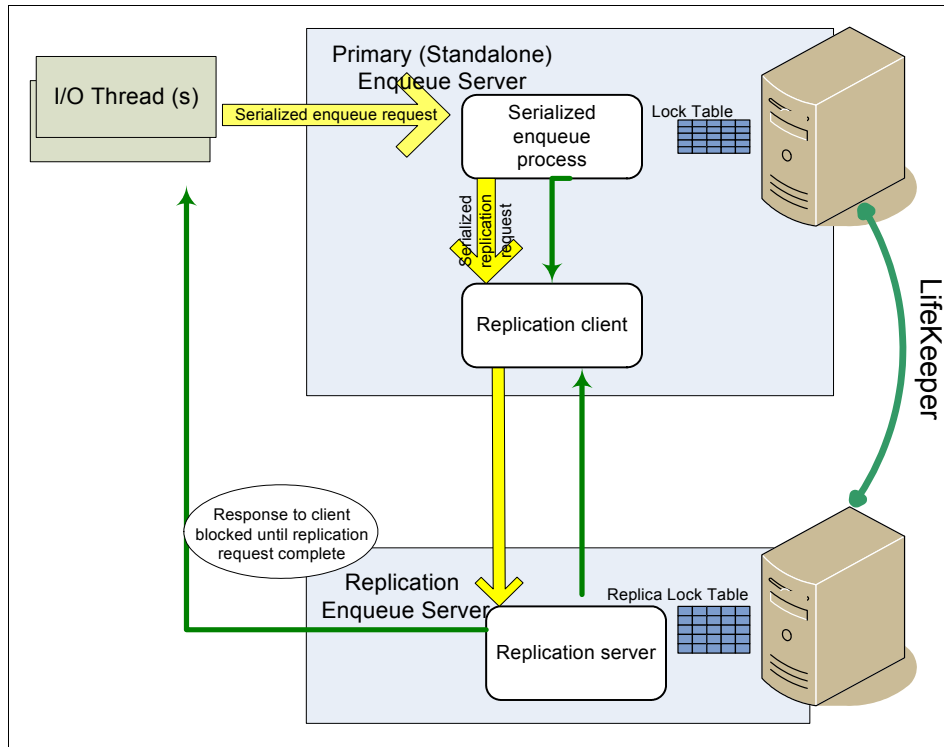


Figure 3-12 Enqueue server high availability

3.7.2 Failure of the database instance

A failure at the database instance also results in an outage. The database instance must be protected with a redundancy technique. In this book, the LifeKeeper DB2 recovery kit is the cluster software protecting the database layer.

If a failure at the database occurs, application threads suffer from a loss of connection. In order for the application to continue its work with minimal interruption, a re-route or reconnect method is required. SAP has the ability to set up a DB reconnect feature. For high availability, this feature in addition to LifeKeeper provides complete protection.

The DB reconnect feature ensures that all work processes of an SAP instance are automatically reconnected to the database as soon as possible after the database instance has been restarted. To the end user, the temporary database failure is almost seamless. The end user is only impacted by the time taken for the database service to be switched over.

3.8 SAP NetWeaver in cluster configurations

There are several different cluster configurations available for protecting the central services instances, database, and file system components for SAP. Based on the workload, the database and central services instances might have to be distributed onto separate servers. SAP also recommends that the database and central services instances be placed into different switchover groups.

Cluster management software, such as LifeKeeper, can be used to protect and automate recovery for each of the SAP components. LifeKeeper can also provide a cluster framework for scaling the Application Server (AS) while providing continuous client connection with cascading failure.

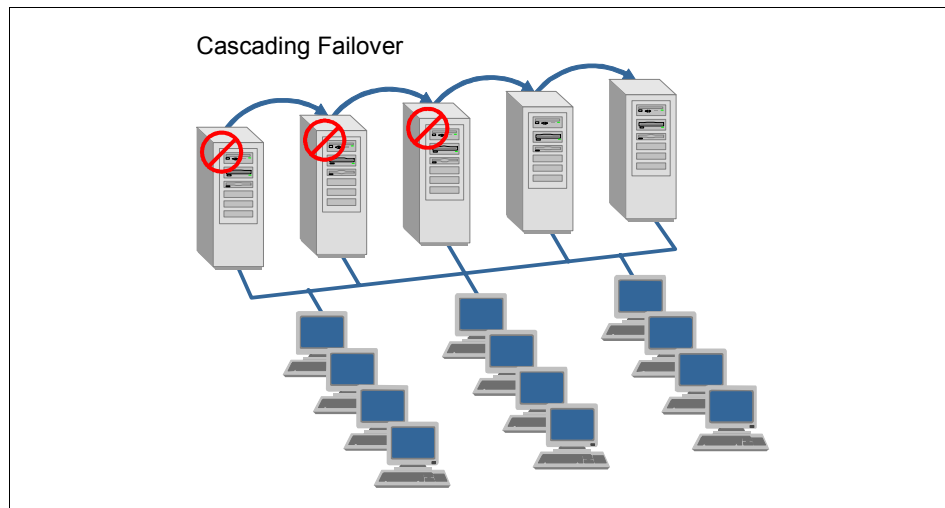


Figure 3-13 Cascading cluster configuration

Providing redundancy of the application server by distribution eliminates the need to perform failover procedures. If an application server fails, user connections can be re-routed to another application server.

For scalability, a hardware load balancer or SAP Web Dispatcher is required to load balance incoming requests. In a high availability architecture, these components should also be implemented with redundancy. Based on security requirements, this redundancy can be implemented either using software clustering or multiple servers.

Because the application server is generally used for scalability and our testing environment was limited to two blades, our configuration did not exploit application servers running on separate servers.

3.8.1 Active/Passive mode

In a two-node Active/Passive configuration, the database instance, central instance, and central services are active on the primary node. The central instance and database instance access files on the shared storage. In the event of a failure, these services can be switched to the standby (passive) node and continue operating with minimal outage.

In an active passive configuration, the secondary (standby server) is idle. Figure 3-14 depicts an Active Passive configuration.

If, for example, a failure occurs at the database layer (DB2), then LifeKeeper first attempts to restart DB2 on the current server (Server A). If this restart is not possible, LifeKeeper switches the DB2 resource, including its dependencies, over to the secondary server (Server B). For this configuration, we recommend that the user manually creates a dependency between that SAP hierarchy and the database hierarchy (the SAP resource must be the parent). In this configuration, SAP and that database would always fail over together.

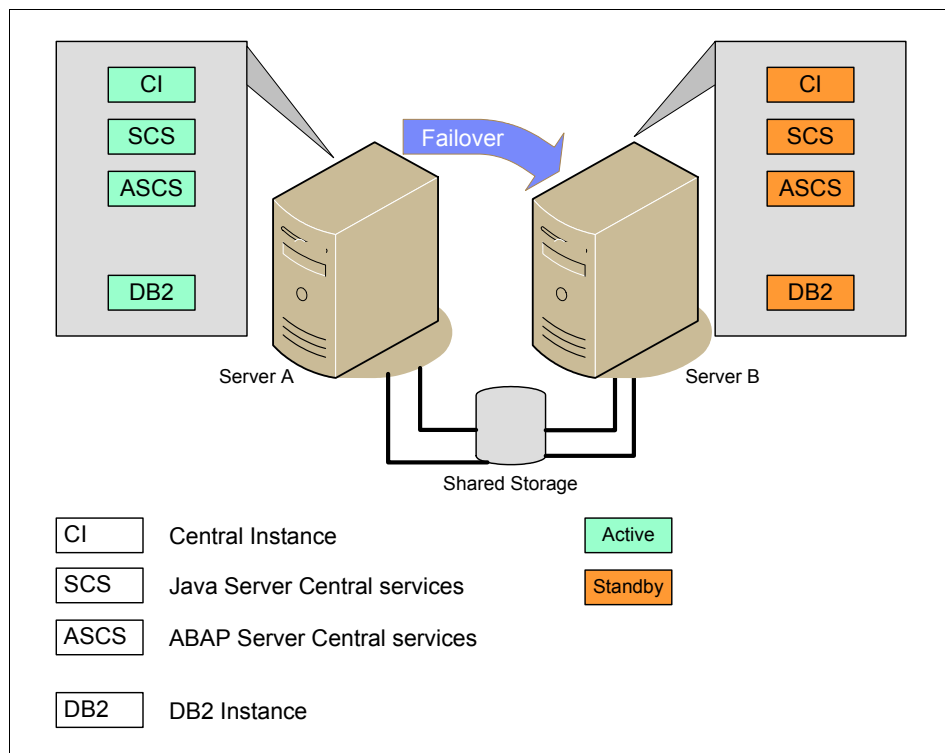


Figure 3-14 Active/Passive mode

3.8.2 Active/Active mode

In a two-node Active/Active configuration, there are active components on each server. Typically, the database instance and central services instances are distributed on different servers to balance the work load.

This means that there is a workload active on each server. It is thus important to size each server appropriately so that it can handle the additional workload, should a failover occur. Figure 3-15 depicts an Active/Active configuration.

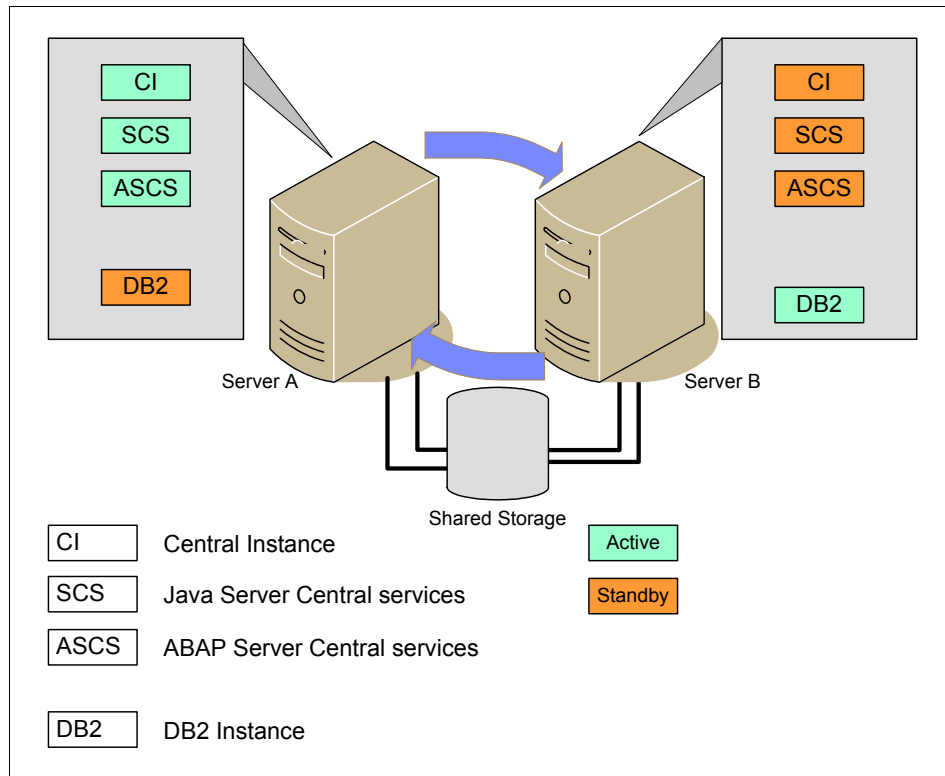


Figure 3-15 Active/Active mode

In both configurations, a shared storage is implemented between the servers. In addition, virtual hostname and IP addresses are configured to allow for independent failover of components.

For example, a configuration can have separate virtual IP addresses for SCS, ASCS, and DB2, each with separate segments on the shared storage to enable independent failover.

In that configuration, if the SCS component fails, the SCS component, including its dependent shared storage, can be switched to the standby server. The virtual hostname and IP address for SCS is taken over by the standby node.

When the ASCS and the SCS are a part of the same SAP system, which is called an ABAP+Java AddIn environment, they must always fail over together.



High availability topologies

In this chapter, we discuss the scenario used to build and test a highly available infrastructure with LifeKeeper for Linux for SAP Netweaver on the SUSE Linux Enterprise Server base operating system. We also include an architectural overview of the test scenario.

The IBM SAP International Competence Center (ISICC) in Walldorf, Germany, provides a rich testing infrastructure that can be used to build complex environments for SAP applications and databases running on different base operating systems.

Information about the IBM SAP International Competence Center in Walldorf can be found at:

<http://www.ibm.com/de/worktogether/customer/isicc.html>

This includes enterprise level IBM equipment such as:

- ▶ IBM System p servers
- ▶ IBM System x servers
- ▶ IBM SAN Volume Controller
- ▶ IBM TotalStorage® DS8000
- ▶ IBM TotalStorage DS4000™

The test scenario for this book was developed using IBM enterprise level technology. This chapter describes how the test scenario for this book was chosen.

The physical topology of a high availability infrastructure is composed of server hardware and communication infrastructure. Along with the server and infrastructure related conditions, we also review the additional software used for and in a high availability topology during the design and architectural phases.

This chapter is intended to assist in making architectural decisions. It covers those conditions deriving from the various elements of the system:

- ▶ “Servers”
- ▶ “Network”
- ▶ “Base operating system”
- ▶ “Cluster software”
- ▶ “Storage”
- ▶ “IBM DB2”
- ▶ “NetWeaver”

4.1 Servers

The physical topology of a high availability infrastructure is composed of server hardware. The server hardware must fulfill requirements given by a base operating system and a cluster management software; it also has to suit the communication infrastructure.

This section explains the elements of a physical server that have to be considered as a single point of failure and gives a summary of the level of redundancy that can be achieved:

- ▶ “Power supply”
- ▶ “Random access memory”
- ▶ “Hard disks”
- ▶ “Local Area Network interfaces”
- ▶ “Storage Area Network interfaces”
- ▶ “Test environment”

The goal of high availability is eliminating single points of failure. As mentioned in Chapter 3., “High availability architectural considerations” on page 19, the physical server itself, as well as the subsystem components may be single points of failure. To establish a high availability environment, at least two servers are required and all communication paths and the power supply have to be redundant to avoid an interruption.

4.1.1 Power supply

Most IBM servers can optionally be equipped with redundant power supplies. Further reference detail is available at <http://www.ibm.com/> by searching for Systems & servers under Products.

For the test scenario in this book, having a redundant power supply was not seen as mandatory because it did not affect any part of the software installation.

4.1.2 Random access memory

Most IBM servers can optionally be equipped with redundant random access. Further reference detail is available at <http://www.ibm.com/> by searching for Systems & servers under Products.

For the test scenario in this book, having redundant random access hard disks was not seen as mandatory because it did not affect any part of the software installation.

4.1.3 Hard disks

Most IBM servers, including Blade servers, can be equipped with at least two internal disks. A disk controller with RAID capability can perform mirroring of the data between these disks. However, in this instance, the disk controller is a single point of failure, but with a lower risk of failure than the disks.

For the test scenario in this book, having redundant random access memory was not seen as mandatory because it did not affect any part of the software installation.

4.1.4 Local Area Network interfaces

Most IBM servers are equipped with at least two Ethernet connectors on their mainboards. Many can be enhanced with additional plug-in cards. The Blades are equipped with two on-board Ethernet connectors. They can be enhanced, either with host bus adapters or network adapters, but not both. For a Blade, this would require an enhancement that uses a second slot in a Blade Center. Because the Blade Center usually only has two built-in independent Ethernet switches, it is acceptable to use just two network adapters on a Blade — more gives no availability advantage.

Depending on the operating system, multiple adapter cards can be used for failover operating or trunking.

In failover mode, network traffic is directed through one of the network adapters and in the event of a failure directed to the other. This operation mode does not require special configuration for a network switch to which the server is attached.

In trunking mode, network traffic is passed to two or more network adapters at the same time. The bandwidth of each single adapter accumulates to the overall bandwidth, so higher throughput is possible. There is no failover, but the total throughput decreases if one adapter fails. This configuration requires special configuration for a network switch to which the server is attached.

Tip: Best practice methodologies suggest having a secondary network adapter on a separate network plug-in card, because a PCI bridge on the mainboard can fail, rendering the on-board network adapters useless. An additional card provides redundancy and enables the communication for the cluster manager to continue.

4.1.5 Storage Area Network interfaces

Host bus adapters provide access to a Storage Area Network. Similar to a Local Area Network, there can be multiple plug-in cards holding Fibre Channel host bus adapters, and a redundant topology would look very similar to an Ethernet. A redundant topology allows a SCSI I/O driver to determine multiple ways to reach different devices. These paths can be used for failover purposes and also as a trunk or load balancer to achieve a bigger throughput. These failover capabilities are part of a driver layer on top of the host bus adapters.

Dual-port host bus adapters can be used, however, there is still a risk that shared components on these adapters might fail, rendering both Storage Area Network interfaces unusable.

4.1.6 Test environment

The following components were chosen for the test scenario:

- ▶ IBM BladeCenter with:
 - IBM Blade servers HS 21
- ▶ IBM TotalStorage DS 8000 storage subsystems, managed using:
 - IBM SAN Volume Controller

The hardware components were located in different server rooms. The server rooms had separated power supply lines to prevent failures by power outages, separate air condition and local uninterrupted power supplies.

The server hardware used consisted of two Blade servers located in two different Blade Centers. The following section gives a short overview of the Blade Center configuration:

- ▶ “Blade Center”
- ▶ “Blade server”
- ▶ “Blade Center Management Module”
- ▶ “Fibre Channel switch”
- ▶ “Ethernet switch”

Blade Center

Each Blade Center contains:

- ▶ Two power supplies
- ▶ Two Qlogic Fibre Channel switches
- ▶ Two Ethernet switches
- ▶ One Management Module



Figure 4-1 IBM Blade Center

The power supply provides redundant electricity for all components in the Blade Center, eliminating the power network as a possible single point of failure. An uninterrupted power supply was put in place in front of one of these power supplies. There was still a risk that the Blade Center would fail, if the uninterrupted power supply failed in the event of a power failure. This risk was accepted for the test scenario.

Each Blade Center is equipped with redundant Qlogic Fibre Channel switches and two Gigabit Ethernet switches to provide redundant connectivity to both the Local Area Network and the Storage Area Network. These connectivity is achieved through an internal back-plane without any cables.

Blade server

A Blade server is a fully featured server supplied as a single unit without power and cooling supplies. The server requires a chassis to provide electric power, cooling and access to a communication infrastructure. This chassis is the Blade Center, and the Blade Centers used for this document have both been equipped with D-Link Gigabit Ethernet switches for Local Area Network access and Qlogic Fibre Channel switches for integration into a Storage Area Network. Figure 4-2 shows a picture of a single Blade server.

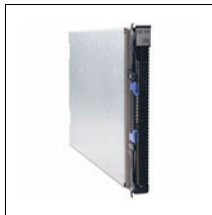


Figure 4-2 IBM Blade server HS 21

Blade Center Management Module

The IBM Blade Center Management Module offers remote management functionality for the Blade servers in a Blade Center. The Blade Center can be accessed through a Web browser via HTTP/HTTPS or through a command line interface using telnet or ssh.

Redundant management modules can be configured in a highly available infrastructure, but it is not typically a requirement. If the remote management functionality is unavailable, the Blade Center can still be administered at the physical console.

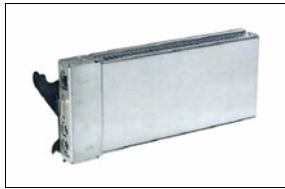


Figure 4-3 A single management module for the IBM Blade Center

The capability to start, stop, monitor status or remotely administer the Blade Center can be sent to a Web browser by forwarding the video output, keyboard, and mouse of the Blade server.

Figure 4-4 is an example display from the Blade Center Management Module.

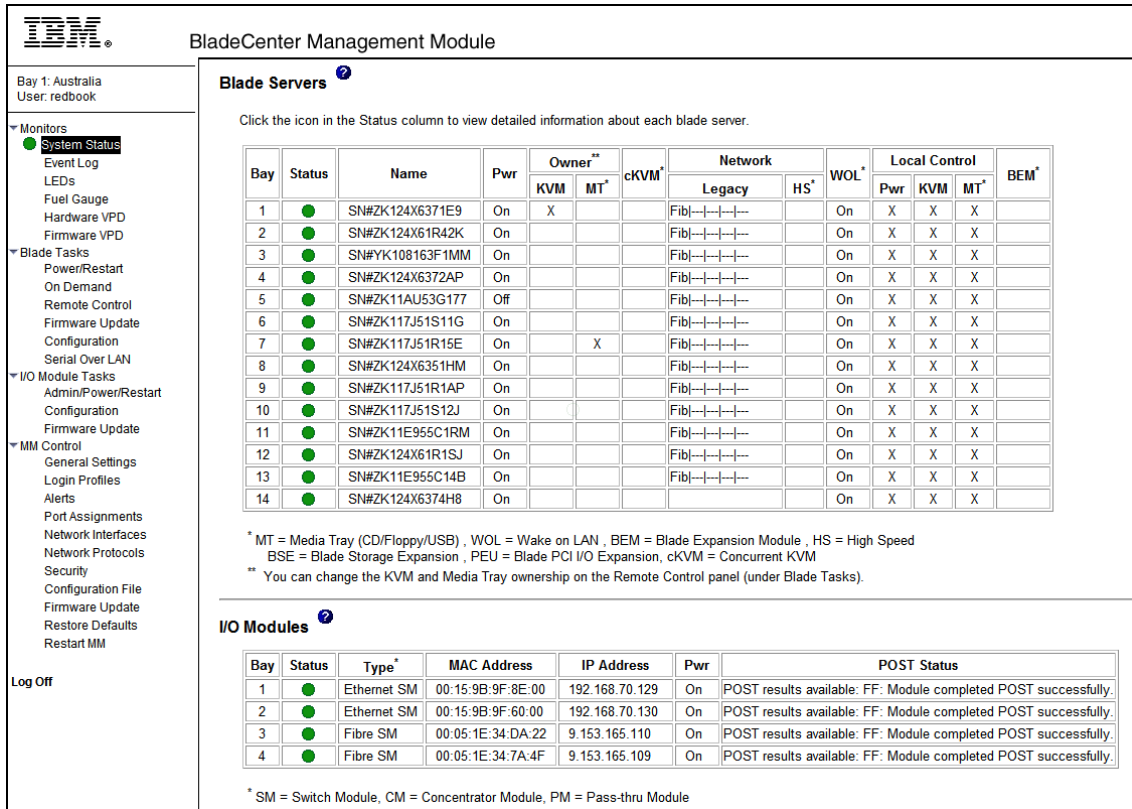


Figure 4-4 A view to the Blade Center Management Module

Fibre Channel switch

The internal Fibre Channel switch of the Blade Center provides connectivity to the Storage Area Network. There is redundant connectivity from each Blade Center to the Storage Area Network.

From an application or operating system perspective, it does not make a difference what type of Fibre Channel switches are used, but it only relates to the Storage Area Network topology and the other components in that network.

In the test scenario used for this book, Qlogic Fibre Channel switches were used.

Ethernet switch

Each blade was equipped with an Ethernet network switch. Though there are different types of switches, it does not generally matter what type of switch is used. If features such as virtual Local Area Network tagging, quality of service routing, or port trunking are required, then an appropriate switch should be used.

For the test scenario in this book, D-Link Ethernet switches were used.

Each server blade was equipped with:

- ▶ Two Intel® Xeon CPUs with a clock frequency of 2.8 GHz
- ▶ 8 GByte of random access memory
- ▶ Two 73 GByte SCSI hard disk drives

Note: These disks were not mirrored for the test scenario. Disk mirroring can work with both a ServeRAID™ controller or Linux Software RAID. There are no implications to the cluster manager software or any of the application.

- ▶ Two network cards
- ▶ Two Qlogic Fibre Channel host bus adapters connecting to the Storage Area Network with a speed of 2 GBit/s

4.2 Network

The Blade centers were connected through Gigabit network providing redundancy and transparency to the hosts. All network adapters of the test Blade servers were in one flat network.

One adapter was configured with an externally routed native host IP address, while the other was configured with a non-routed private IP address during the base operating system installation. So both Blade servers could be reached on one interface (the first interface) from outside while they have only have to reach each other on the second interface.

4.3 Base operating system

This section discusses what influence the base operating system has on the physical topology.

4.3.1 Linux operating system

SUSE Linux Enterprise Server 10 base operating system allows the selection of *SAP Application Server base* installation package for easy pre-configured installation that can be used with SAP software.

This section of the book only covers those aspects of the operating system configurations that are relevant for the cluster setup and differs from a default installation.

The SUSE Linux Enterprise Server base operating provides built-in high availability features such as:

- ▶ “Redundant Local Area Network connection”
- ▶ “Redundant Storage Area Network connection”
- ▶ “Mirroring across storage subsystems”
- ▶ “Logical Volume Manager”
- ▶ “File systems”
- ▶ “High availability storage topology”

Redundant Local Area Network connection

The bonding feature allows use of two or more network adapters for different configurations. Some of the features support:

- ▶ Active-backup configuration
- ▶ Link monitoring and active ARP-based monitoring
- ▶ Weighted load-balancing through multiple adapters

The bonding module creates a layer over several physical network interfaces.

While the physical interfaces are usually called eth0, eth1, eth2, and so on, the virtual layer for bonding is called bond0, bond1, and so on.

In active-backup configuration, one interface is used while the other is on standby. When the active interface stops working, the network traffic is directed through the stand-by or backup interface.

Example 4-1 shows a configuration for a virtual interface bond0 consisting of an active interface eth0 and a secondary and stand-by interface eth2. These are configured in active-backup mode to provide protection in the event of a failure.

Example 4-1 Ethernet bonding configuration

Ethernet Channel Bonding Driver: v2.6.5 (November 4, 2005)

```
Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: eth0
MII Status: up
MII Polling Interval (ms): 0
Up Delay (ms): 0
Down Delay (ms): 0
```

Slave Interface: eth0
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:0e:50:1b:e6:36

Slave Interface: eth2
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:08:ef:0c:c9:ef

Redundant Storage Area Network connection

The Linux standard kernel includes code for dealing with devices that are recognized through different paths. This is a part of the device mapper driver layer and a module called Device-Mapper Multipath I/O (dm-mpio).

The Linux manual page `multipath(8)`, shown with the command `man multipath`, says: *“multipath is used to detect multiple paths to devices for failover or performance reasons and coalesces them”*.

Device-Mapper Multipath I/O provides a driver layer that links several paths recognized as individual block devices to one logical device providing access across these paths through different policies. This can be used for load balancing and failover purposes. This driver layer is a generic framework for multipath access to various storage subsystems.

In Linux, the individual paths to one and the same storage volume on a Storage Area Network appear each as a single disk drive. These are grouped by the multipath driver layer.

Note: The device mapper multipath is the preferred way to realize multipath Storage Area Network connections. While similar functionality is still built in the Software RAID module (md), it is no longer supported. Current development occurs in the Device-Mapper Multipath I/O module and it has a broad support from many Enterprise Linux distributors and hardware vendors.

The Device-Mapper Multipath I/O module includes an interface for external programs that do vendor specific interaction with a storage subsystem. These modules are called personalities, and these personalities exist, for example, for:

- ▶ IBM SAN Volume Controller
- ▶ IBM DS 8000
- ▶ IBM DS 6000
- ▶ IBM DS 4000
- ▶ IBM System Storage™ N Series

These provide access to storage subsystems from other vendors and can also be used in a heterogeneous environment.

Support for a particular configuration has to be verified when doing the setup. The device mapper multipath I/O personality for DS 8000 is supported by IBM TotalStorage and replacing the sub-system device driver (SDD). At the time of writing, the only supported driver for the DS 4000 series is the IBM RDAC driver.

The SUSE Linux Enterprise Server distribution provides enterprise support for many storage subsystems. Since supported configurations can change due to normal product life cycle, review the hardware support matrix and pertinent information on the Novell support Web site. Device-Mapper Multipath I/O support should be reviewed for the particular storage subsystem. The Novell document 3203179 shows the latest list of supported storage subsystems at:

http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=3203179&sliceId=SAL_Public&dialogID=58410954&stateId=0%200%2058414650

Important: For a high availability cluster, the configuration for the Device-Mapper Multipath I/O driver layer has to be equal. The file `/etc/multipath.conf` must be the same on all cluster nodes, and the multipath services have to be enabled.

Mirroring across storage subsystems

The Linux operating system provides support for mirroring data across disk drives. This mirroring is performed on block level. The mirroring works on multiple physical disk drives and provides a virtual disk. This virtual disk provides block level access for other disk I/O related layers such as the Logical Volume Manager or a file system. The Software RAID driver includes RAID levels 0, 1, 5, 6 and 10. The Software RAID driver allows hot swapping of disks, assuming that the hardware allows and supports it.

The Software RAID module works on block devices and provides another virtual block device for further usage. Software RAID can be used with:

- ▶ Entire disk drives
- ▶ Partitions
- ▶ Virtual devices provided by the Device-mapper Multipath I/O module

The virtual block devices provided by Software RAID are called `/dev/md0`, `/dev/md1` and so on, and they can be used directly, without having a partition table. For example, they can be used with **pvcreate** to prepare them for usage in the Logical Volume Manager. Figure 4-5 shows an example of the Software RAID layer:

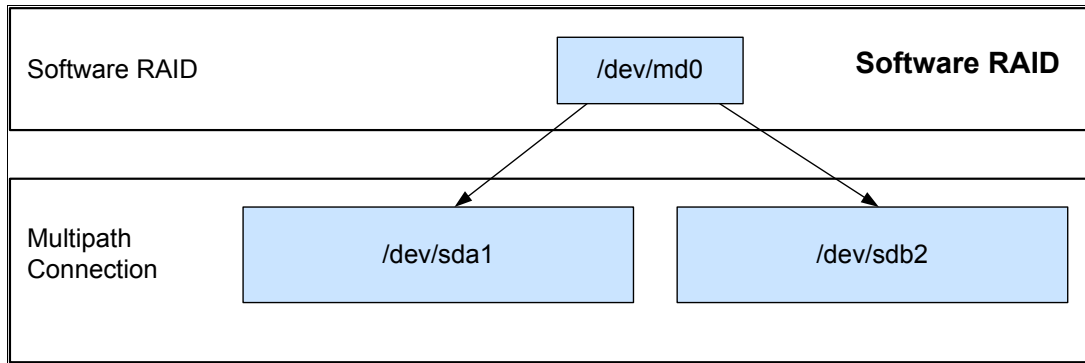


Figure 4-5 Example for disk mirroring with the Software RAID driver module

Attention: The Software RAID module provides multipath functionality. While this has been supported by several vendors in the past and is known to work in general, it is no longer supported in the latest Enterprise Linux distributions and is not supported by the distributors or hardware vendors.

Multipath support for Linux is achieved through the device-mapper multipath module because it provides support for a broad range of storage subsystems from most vendors, and is supported by Linux distributors and hardware vendors.

Vendor-specific drivers might be required. For example, for the IBM DS 4xxx series, the IBM RDAC driver software must be used.

For a high availability cluster, the configuration for the Software RAID module has to be identical on each node. The file `/etc/mdadm.conf` must be identical and automatic startup of RAID devices must be disabled on each node.

The `mdadm` daemon provides monitoring and e-mail notification for Software RAID array events and alerts and should normally be enabled. As LifeKeeper monitors these RAID arrays, it is no longer necessary to use `mdadm`. A running `mdadm` is disabled by LifeKeeper. If `mdadm` should run additionally, it can be integrated into the LifeKeeper resource hierarchy. An example is given in 7.3.2, “Changing LifeKeeper configuration values” on page 256.

It is possible to use mirroring for both the internal disks and shared disks, concurrently. However, it can lead to a conflict if automatic startup of RAID devices has to be turned off. If this is a requirement, the Storage Area Network host bus adapter can be loaded at a later time. After a standard SUSE Linux Enterprise Server installation, the host bus adapter drives are loaded from within the initial RAM disk. These can be moved to a later step in the boot process.

Attention: If the automatic startup of RAID devices is not disabled, this can lead to a situation where multiple cluster nodes use the same data at the same time, without any cluster control. This can end up in loss of all data.

Logical Volume Manager

The Linux operating system provides a Logical Volume Manager. The Logical Volume Manager gives more flexibility, allows quick enhancement of volumes, and supports easier storage administration on a Linux server.

In a high availability environment, using the Logical Volume Manager allows easy storage enhancement, depending on the file system or the application used, even while it is online.

Attention: It is possible to increase the size of a volume in a storage subsystem. At the time of writing this book, the Linux operating system was unable to recognize the increase. Currently, a LUN must be added and the volume extended using the Logical Volume Manager. Refer to the latest version of the code for the Storage Area Network and resizing of logical volume details.

If necessary, the Logical Volume Manager can even be used for striping, known as distributed parallel writing to multiple disks. This can be useful for some storage subsystems that cannot provide striping themselves.

Tip: With both Software RAID or the Logical Volume Manager, it is possible to use an entire physical disk. Unfortunately, a couple of tools do not recognize that a disk is used by another driver layer. Therefore, it is a good idea to use partition entries for physical disks, to prevent data from being overwritten accidentally.

In this book, the Logical Volume Manager was used for flexibility and to demonstrate integration with the LifeKeeper for Linux software.

Tip: When using the Logical Volume Manager between file systems and disk I/O, it is possible to move data from one disk array to another, or even to migrate it from one storage subsystem to another, while it can be used from an application at the same time.

Figure 4-6 shows an example volume group, consisting of three different disk drives and three logical volumes used for file systems.

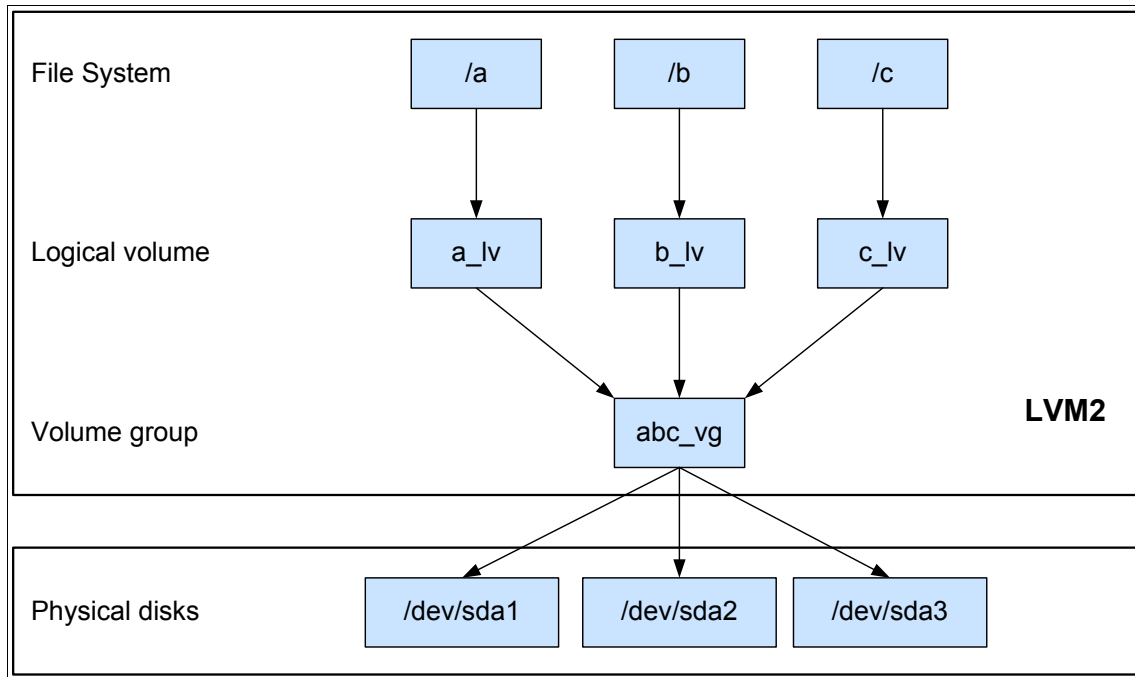


Figure 4-6 Example for a volume group with the Linux Logical Volume Manager

Note: This driver chain had locking issues in the past. Always use the latest version of this software. This chain was built and tested with SUSE Linux Enterprise Server 10 Service Pack 1.

File systems

The SUSE Linux Enterprise Server base operating systems support these file systems:

- ▶ ReiserFS v3
- ▶ ext3
- ▶ XFS
- ▶ OCFS2

For further detail, see:

<http://www.novell.com/linux/filesystems/faq.html>

Using two different types of file systems for shared storage in a cluster environment and for those file systems on internal disks even allows a failure of the file system code without affecting the cluster node. The Linux kernel might eject a kernel module if it shows an error, and that could happen to a file system module too. So if a shared storage is running with the same file systems as the internal disks, a problem with the shared storage can render the internal disk inaccessible. This leads to unpredictable results.

It was decided to use XFS for the internal disks based on best practice recommendations as follows:

- ▶ XFS can be enhanced online.
- ▶ XFS is a journaling file system with good performance.

The ext3 file system was used for the shared storage for the following reasons:

- ▶ ext3 is a journaling file system.
- ▶ ext3 is known to have best performance together with database files.
- ▶ Although ext3 cannot be resized while online with SUSE Linux Enterprise Server, a database for SAP can be increased dynamically through adding another logical volume with a new file system.
- ▶ An ext3 file system can be enhanced while it is unmounted.

There were no special requirements for the disks holding the operating system, and the file system layout for the application and database is preset by the SAP installation.

High availability storage topology

To achieve a highly available and flexible shared storage, different driver layers can be combined as follows:

- ▶ Device mapper multipath I/O for redundant storage connectivity
- ▶ Host based mirroring with Software RAID for redundant shared storage
- ▶ Logical Volume Manager for flexibility

Figure 4-7 shows the stack of all these layers used together.

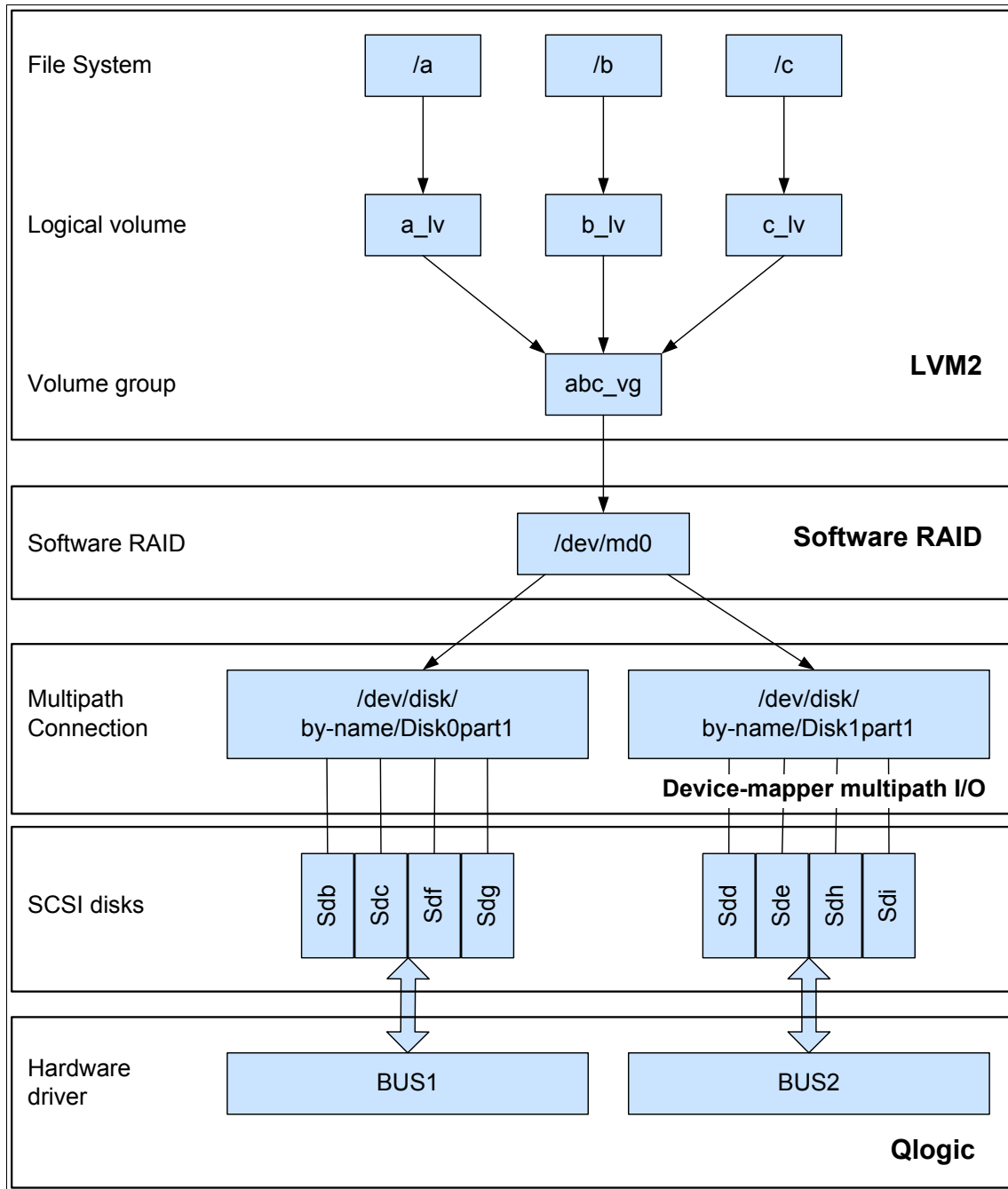


Figure 4-7 Overview of the storage related Linux driver stack

4.4 Cluster software

Many high availability clusters consist of one active server running an application and a stand-by or passive server that is idle and waiting. In this case, the resources on that node are just wasted, and if the passive node is not used, it is possible that the hardware problems do not get recognized before a failover happens. It would be better to share some load across the servers by forming a cluster. For a better overall performance, it is even possible to split application components across multiple servers and benefit from adding up the power of several servers.

LifeKeeper for Linux supports cluster configurations with multiple servers. While a cluster can consist of many nodes, it is possible to have some applications and other resources bound to only a few of them. They need to have priorities assigned so that an application or resource runs preferably on a primary node, but can be taken over by another, and if that one fails, by a further one. So, in case of some bad failure, an application would still be available, probably with limited performance, but keeping a business application running.

An active/passive scenario would just require:

- ▶ One service IP address
- ▶ One shared storage volume
- ▶ One common software installation

An active/active scenario would require at least:

- ▶ Two IP addresses
- ▶ Two shared storage volumes.

The two Blade servers stay independent, but have access to a common Local Area Network and common resources on the Storage Area Network. Resources such as cluster IP addresses and shared storage are managed through the LifeKeeper for Linux cluster manager only.

Note: The test scenario in this book was started with an active/passive configuration that could be modified to an active/active configuration.

Therefore, the topology was split with NetWeaver software and the IBM DB2 database software on two different nodes in a cluster.

This required:

- ▶ Two individual service IP addresses
- ▶ Two individual pieces of shared storage

As it was planned to have the shared storage data mirrored on two storage sub-systems, it became necessary to have:

- ▶ Two storage volumes of the same size for the mirrored NetWeaver data
- ▶ Two storage volumes of the same size for the mirrored IBM DB2 data

The LifeKeeper for Linux software can work with any layout on a shared storage. It does not have special requirements because it does not keep any data on a shared storage itself.

The LifeKeeper for Linux cluster manager software can integrate the full stack of:

- ▶ File systems
- ▶ Logical Volume Manager
- ▶ Software RAID
- ▶ Multi-path Storage Area Network connectivity
- ▶ Individual Storage Area Network volumes

It was decided to use this stack of drivers to test and prove the integration with LifeKeeper for Linux. Storage based data replication was not considered because LifeKeeper for Linux does not take advantage of it.

A storage network and storage subsystem used by LifeKeeper has to fulfill requirements for SCSI locking. Only certified storage subsystems can be used with LifeKeeper for Linux.

The shared storage resides on a Storage Area Network based in IBM TotalStorage DS 8000 storage subsystems with the IBM SAN Volume Controller as a layer for virtualization and management. This reflects a common and typical enterprise storage environment.

It has been verified that the full driver chain as mentioned in Figure 4-7 is supported by the LifeKeeper for Linux cluster software. Application Recovery Kit covering all the layers are available from SteelEye.

Note: The SteelEye LifeKeeper software supports cluster-internal communication (heartbeat) through a serial line interface, named TTY communication path. As it is independent from third-party components such as a network, it is a good idea to use it when possible.

4.5 Storage

This section covers two aspects of the shared storage:

- ▶ “Storage scenario”
- ▶ “Storage layout”

The storage scenario describes the Storage Area Network infrastructure, and the storage layout describes how it was configured and used from the cluster servers.

4.5.1 Storage scenario

As the NetWeaver binary software components cannot reside on local storage volumes, both NetWeaver and the IBM DB2 binaries were placed onto their shared storage volumes for consistency.

On each storage subsystem, two volumes were created through the SAN Volume Controller, and access to these volumes was granted through both Storage Area Network interfaces of the Blade servers.

This topology was prepared with the idea of having an active/active resource configuration in the cluster. The SAP application data and the database data were put on different shared volumes. For host based mirroring, one of these volumes had to reside on each of the storage subsystems. So, two pairs of equal sized volumes were created on the Storage Area Network.

Mirrors, physical volumes, volume groups, logical volumes, and file systems have to be set up manually before the software installation can be performed, but automatically integrated into the resource hierarchy of the cluster manager.

Figure 4-8 shows an overview of the storage topology used in the test scenario, including the IBM SAN Volume Controller as the storage virtualization layer.

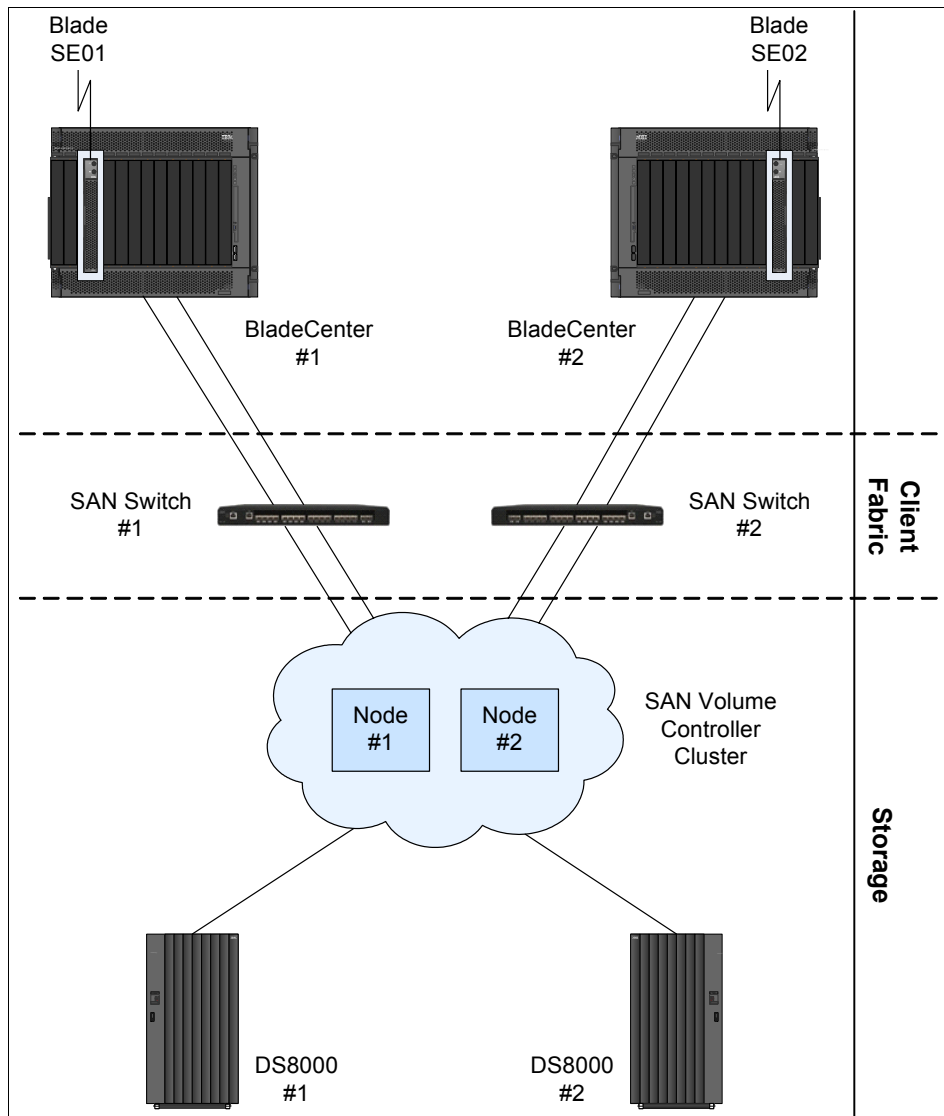


Figure 4-8 Storage topology in the test scenario

4.5.2 Storage layout

This section covers:

- ▶ “NetWeaver storage prerequisites”,
- ▶ “DB2 LUW storage prerequisites”
- ▶ “Storage configuration”

NetWeaver storage prerequisites

The NetWeaver application requires a few specific file systems. It is possible to create all these file systems inside one volume group using the Logical Volume Manager.

DB2 LUW storage prerequisites

The DB2 LUW database requires a number of file systems for installation. For performance, these file systems are typically split across different physical disks on a storage subsystem.

For the test scenario, it was acceptable to put all file systems on one volume group with a focus on availability and integration. This design was not implemented for performance.

Storage configuration

Figure 4-9 shows the full storage configuration, including the affected driver layers, such as:

- ▶ File system
- ▶ Logical Volume Manager
- ▶ Software RAID
- ▶ Device-mapper Multipath I/O
- ▶ SCSI driver layer

The physical topology in the Storage Area Network has no influence on functionality and failover tests, but can be considered for performance optimization.

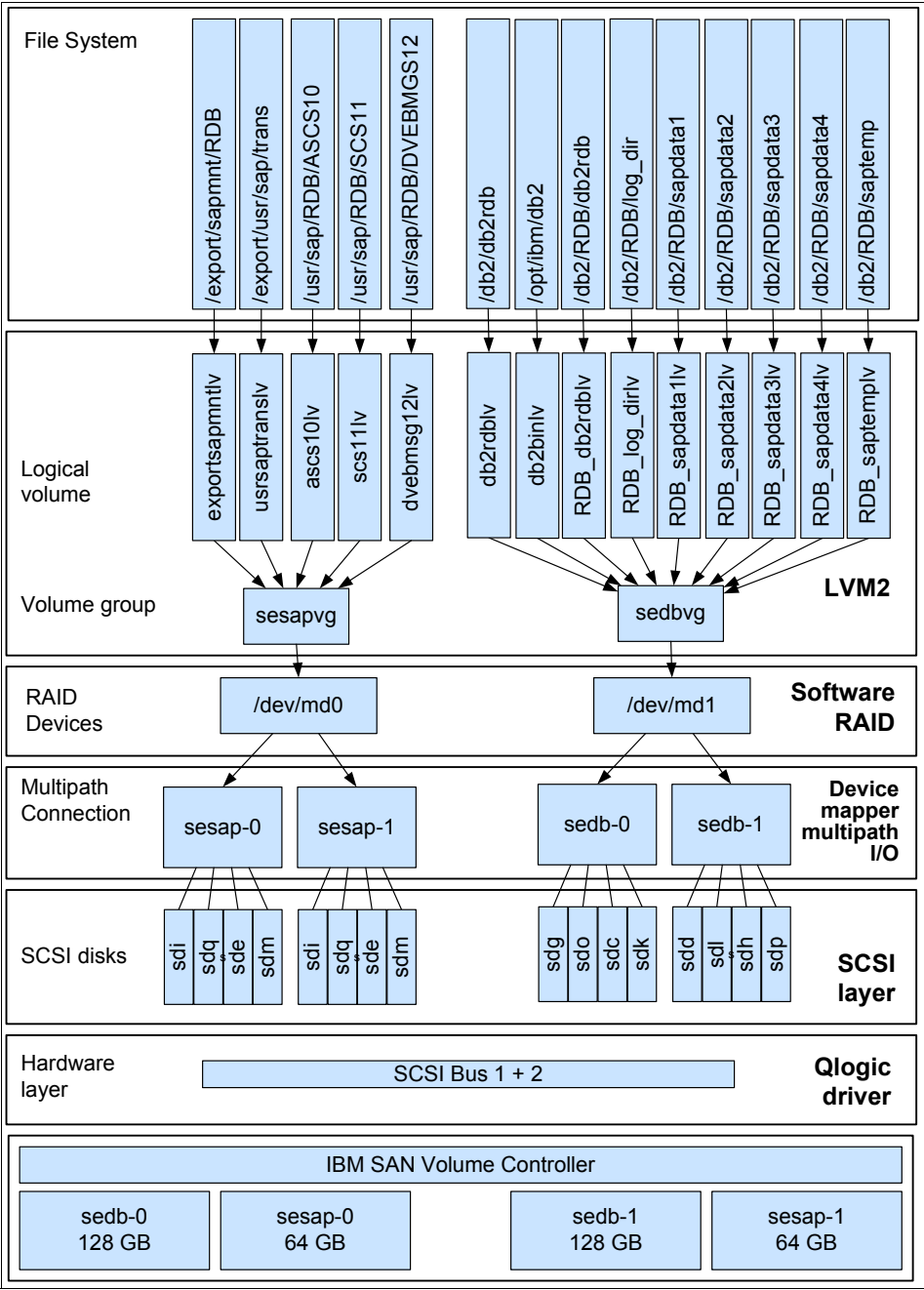


Figure 4-9 Storage configuration for the test scenario

4.6 IBM DB2

The IBM DB2 relational database management system software binaries have to be installed before the SAP installation can be performed. The SAP installer automatically installs and configures the instance and database according to SAP's requirements. In this book, the database was installed and configured for a clustered environment.

The installation directory for the binaries and the database home directory structure can be placed in separate locations. When a DB2 instance is created, soft links are created under the instance home in the sqllib directory to point to the installation directory of the binaries. In a clustered environment, the database home directory must reside on shared storage.

Because it was a requirement that the NetWeaver binaries be placed on shared storage, for consistency, the DB2 binaries were also placed on shared storage. This was not implemented as a requirement from LifeKeeper for Linux, but rather to maintain consistency with the installation components.

In a Linux environment, it is also necessary to ensure client connectivity is available on remote nodes. For Windows, SAP also supports the zero footprint client. The zero footprint client can be distributed centrally, so client software installations on each application server are no longer required.

In a cluster topology, a service IP address for DB2 is required for client service and must be configured. In the event of a failure, both the service IP address and the shared storage must be switched over to the standby server. The service IP address enables client connections to reconnect to the database.

Some commands, such as the **restart** option on **db2start**, use remote command execution. In addition, if the DB2 data partitioning feature (DPF) is used, instance commands must be executed at all partitions in the `db2nodes.cfg`. Some other examples of remotely executed commands are **db2_a11** and **rah** utilities.

These commands are executed by default with remote shell (rsh). Since the remote shell utility sends data and password information in an unencrypted format, this is vulnerable and considered to be insecure. DB2 also provides the ability to configure secure shell as an alternative. DB2 provides support for either public key authentication or host based authentication. To ensure secure shell functions properly, the instance owner id should be setup using trusted hosts and thus, avoid prompting on any participating host.

Specifics on how to configure DB2 with secure shell are documented in section 5.4.4, “Configuring DB2 settings after SAPinst” on page 135.

Figure 4-10 shows a logical overview of the whole scenario, including the DB2 database installation.

4.7 NetWeaver

For the test environment setup, the following SAP recommendations were observed:

- ▶ The utilization of the enqueue replication server both for the ABAP and Java enqueue servers to provide resilience to these services
- ▶ An additional application server, besides the central instance, which was installed to provide an additional layer of redundancy and scalability
- ▶ Creation of distinct resource groups in the cluster management software separating database (longer recovery time) and central services
- ▶ Inclusion of AS (central instance and add-in instance) in a switchover group

For the SAP system installed in the test environment, the name for both the database SID and SAP SID was RDB.

4.7.1 NetWeaver 7.0 components

In the test scenario for this book, the components were distributed and named as shown in Figure 4-10.

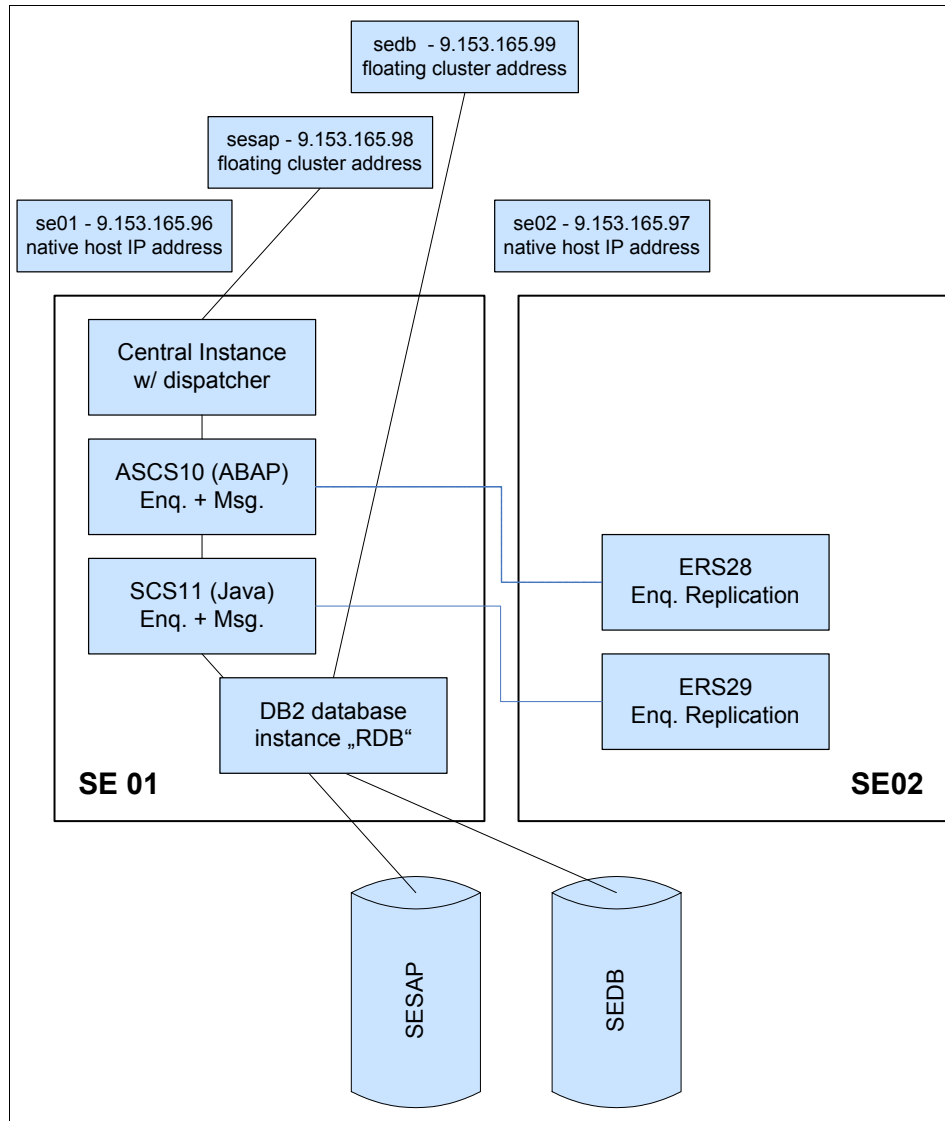


Figure 4-10 Logical view to the application components

This is the distribution for an active-passive scenario. For testing purposes in this book, the application server and the enqueue replication servers were installed on the se02 server. However, for a production system, these components can be installed on a third server.

Regarding the SAP instances:

- ▶ Central instance (DVEBMGS12)
- ▶ ABAP central services (ASCS10)
- ▶ Java central services (SCS11)
- ▶ Application server (D15)
- ▶ Enqueue replication server (ERS28 and ERS29)

Each one of them has a set of services or servers as listed in Table 4-1.

Table 4-1 SAP RDB system instances of the test environment

Instance	Description
ASCS10	The central services instances were separated from the central instances in order to concentrate the single point of failure components: <ul style="list-style-type: none">▶ ABAP enqueue server▶ ABAP message server
SCS11	Like ASCS, the Java instance concentrates the similar services: <ul style="list-style-type: none">▶ Java enqueue server▶ Java message server
DVEBMGS12	DVEBMGS12 is the central instance of the RDB SAP system. The single points of failure are no longer in this instance, they were moved to ASCS instance. The services here are now: <ul style="list-style-type: none">▶ Dialog work process▶ Update work process▶ Batch work process▶ Spool work process▶ Gateway
ERS28	The enqueue replication server ensures that a backup copy of lock table is available in case of failure: <ul style="list-style-type: none">▶ ABAP enqueue replication
ERS29	The enqueue replication server ensures that a backup copy of lock table is available in case of failure: <ul style="list-style-type: none">▶ Java enqueue replication
D15	The application server ensures that the services provided by the central instance are replicated, avoiding the need to switch over to it in case of failures: <ul style="list-style-type: none">▶ Dialog work process▶ Update work process▶ Batch work process▶ Spool work process▶ Gateway

In NetWeaver 7.0, the software components are grouped to provide an easy way to distribute them in a high available environment.

Components that could not be replicated were grouped together in a Server Central Services instance. This distribution model was already present in Java Application Server and was extended to ABAP Application as well. Usually these components cannot be replicated because they must be unique in an SAP system.

There are other components that cannot have multiple occurrences (replicated) and can only exist once within an SAP system; thus, they are a single point of failure. The SAP central filesystem and the database are examples.

4.7.2 Protecting the points of failure

During the distribution of the software components between the cluster groups, it is important to keep the resources as small as possible. This avoids longer recovery times. SAP recommends that the single points of failure be grouped as shown in Figure 4-11.

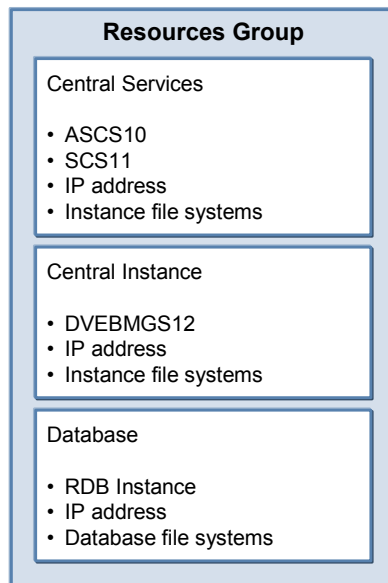


Figure 4-11 Distribution of services in switchover groups

The components in the Central Instance of NetWeaver 7.0 can be replicated, adding an Application Server to the high available system. In our test environment, an application server was installed in order to provide the necessary redundancy for the replicable components. For a highly available architecture, it is important to provide redundancy for the application server. In production systems, there may be more than two application servers to support scalability, but in an HA architecture, a minimal of two are recommended.

In order to minimize switchover requirements, the replicable SAP components (Application Server), are distributed in different servers. When a failure occurs in one application server, only the users connected to that application are affected with the loss of their sessions. The affected users can connect again to an available application server. For the Java usage type, sessions are available even in a failover situation. Further detail in NetWeaver Technical Infrastructure is available at:

<https://service.sap.com/installnw7>

LifeKeeper was used to control the behavior of the central instance and additional application servers in the event of failure. Further details about the failover scenarios tested can be found in Chapter 6., “Testing and failover scenarios” on page 197.

For the single points of failure in the SAP system, again LifeKeeper was used to provide the ability to detect a fail and restart or if necessary switch the failed component from one node to other. The ABAP central services and Java central services were configured as a single resource in the cluster and when necessary they are switched from the active host to the standby one.

During the switch of central services (ASCS and SCS) to another server, the SAP system lose their ABAP and Java message servers, as well as their ABAP and Java enqueue servers. Figure 4-12 shows the distribution of components in between the hosts:

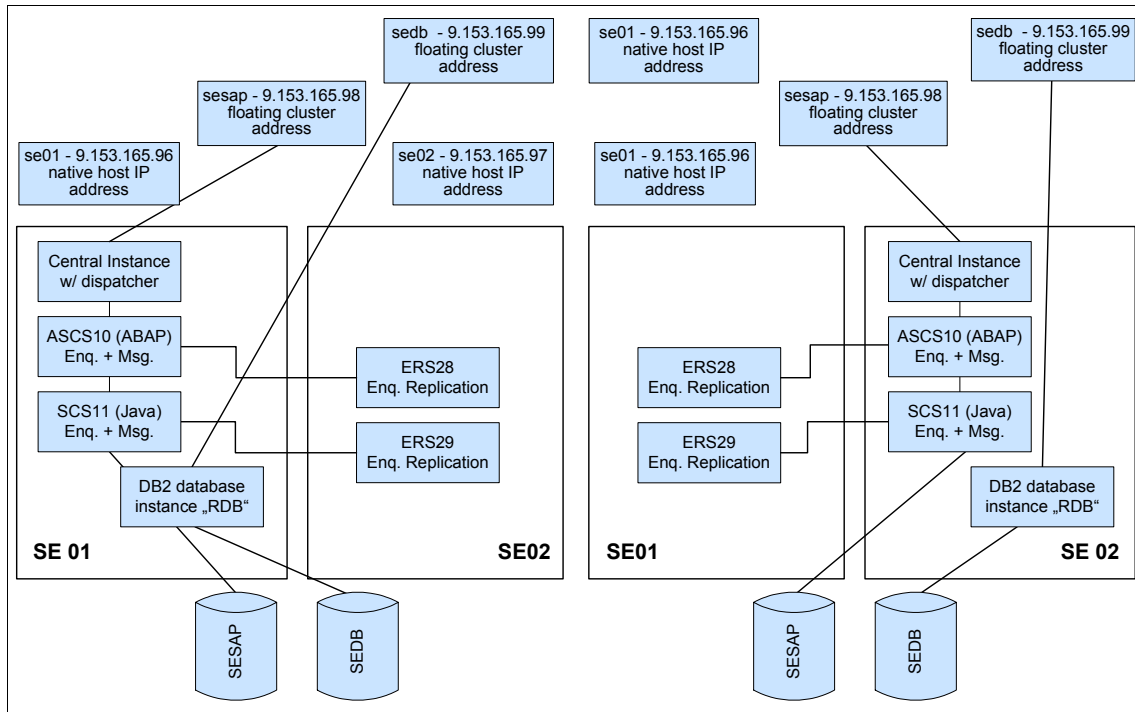


Figure 4-12 Logical view of a fail-over situation

Even if the failover time is minimized, the affects of an outage on a production system can be expensive because the locks held by the application are lost and the transactions are reset.

To prevent this, SAP created the enqueue replication server, an instance in which the function is a replicate of the content of the enqueue server. The enqueue replication server keeps a copy of the enqueue server table. When an enqueue server restarts, it uses the enqueue replication server copy to rebuild the enqueue table.

In the test environment, two active enqueue servers and two standby servers were set up, two for ABAP and two for Java. The replication enqueue servers were always active in the server where the enqueue server was not running to provide the necessary redundancy. LifeKeeper monitored the state of the enqueue server to take the appropriate action in a event of failure.

4.7.3 Database connectivity

In the event of database failure or any other database unavailability, the SAP system does not have to be restarted because the ABAP and the Java processes can automatically reconnect.

After a database failure is recognized, the DB Reconnect feature of SAP attempts to reconnect to that database. A SAP system is unable to operate without the database and therefore stops, if it fails to reconnect.

To prevent this, SAP created the enqueue replication server. The enqueue replication server keeps a copy of the enqueue server table. When an enqueue server restarts, it uses the copy of the table held by the enqueue replication server to rebuild the enqueue table.

The failed database can be restarted on the same server or on the standby server. The DB Reconnect feature then ensures that the SAP system and all the applications are reconnected.

4.7.4 SAP file systems

The SAP file system is a single point of failure because it is shared among all instances. In the test scenario, its protection is achieved by including it as a resource under the SAP system group. Table 4-2 shows the SAP file system as well its initial state and a brief explanation of what is keeping there.

Table 4-2 Shared file systems

File System	Size (MB)	Volume Group	Permissions	Owner:Group
/export/sapmnt/RDB	2048	sesapvg	775	rdbadm:sapsys
/usr/sap/RDB/DVEBMGS10	1024	sesapvg	775	rdbadm:sapsys
/usr/sap/RDB/SCS11	1024	sesapvg	775	rdbadm:sapsys
/usr/sap/RDB/ASCS10	512	sesapvg	775	rdbadm:sapsys
/export/usr/sap/trans	512	sesapvg	775	rdbadm:sapsys
/db2/db2rdb	1792	sedbvg	755	db2rdb:db2rdbadm
/db2/RDB/log_dir	1536	sedbvg	755	db2rdb:db2rdbadm
/db2/RDB/db2dump	128	sedbvg	750	db2rdb:db2rdbadm
/db2/RDB/db2rdb	256	sedbvg	750	db2rdb:db2rdbadm

File System	Size (MB)	Volume Group	Permissions	Owner:Group
/db2/RDB/saptemp1	1024	sedbvg	750	db2rdb:db2rdbadm
/db2/RDB/sapdata1	30720	sedbvg	750	db2rdb:db2rdbadm
/db2/RDB/sapdata2	10240	sedbvg	750	db2rdb:db2rdbadm
/db2/RDB/sapdata3	10240	sedbvg	750	db2rdb:db2rdbadm
/db2/RDB/sapdata4	10240	sedbvg	750	db2rdb:db2rdbadm



High availability implementation

In this chapter, we illustrate the installation of SAP NetWeaver implementation using LifeKeeper on Novell SUSE Linux Enterprise Server for high availability.

We cover the following topics:

- ▶ “Prerequisites”
- ▶ “Base operating system installation”
- ▶ “Shared storage”
- ▶ “DB2 Linux, UNIX, and Windows Enterprise Server installation”
- ▶ “SAP NetWeaver installation”
- ▶ “LifeKeeper cluster software installation”
- ▶ “Cluster configuration”
- ▶ “Creating resources and hierarchies to protect applications”

5.1 Prerequisites

This section describes the prerequisites for the installation of SAP NetWeaver with LifeKeeper over Novell SUSE Linux Enterprise Server. This can change over time, and we recommend that you review all current documentation for accuracy.

5.1.1 Hardware

Because prerequisites and supported configurations can change due to normal variations in product life cycle, we recommend that you review all current documentation for accuracy.

The installation created in this book was based on IBM Blade server technology and capabilities. To determine workload capacity for a server, a sizing estimate should be performed for SAP NetWeaver.

This sizing process has different methodologies and requires participation from both functional and technical areas within the organization. The team has to be involved in determining and understanding the initial business requirements, such as the functional response time required. Based on this data, an estimate can be determined using the “T-shirt” methodology (small, medium, large, extra large), and iterations can evolve during the project for accuracy.

Figure 5-1 shows the sizing methods and indicates when they can be used.

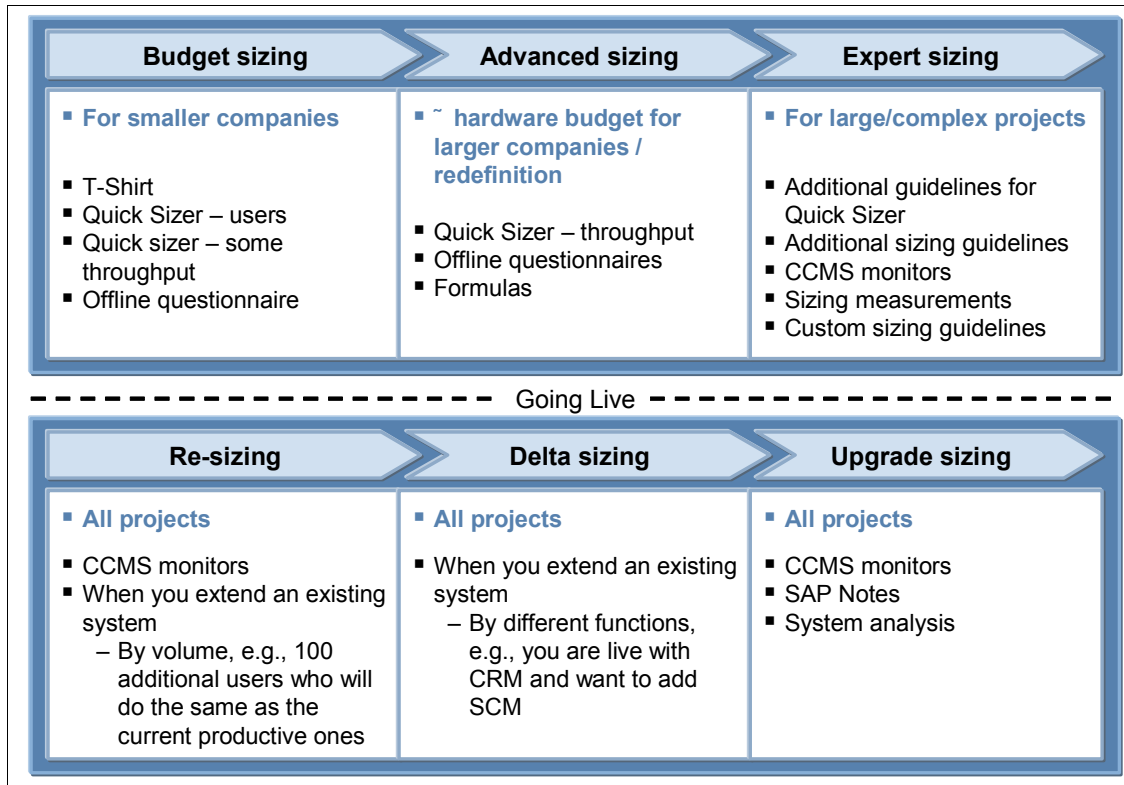


Figure 5-1 Sizing methods and relation with the project phase

On an existing installation, this process is called re-sizing and has to include the actual usage of the installation and the new requirements. This process usually includes filling forms to reflect the current usage of the servers, the capacity level installed, and the new requirements. The IBM SAP Competence Center provides a tool, called IBM Insight for SAP, to help with this process.

The IBM Insight for the SAP utility program and its subsequent reporting and analysis process provide a convenient, high-level, workload analysis for in-production SAP system landscapes. This service and utility are provided free of charge by the IBM America's Advanced Technical Support (ATS) and America's Techline organizations.

The Insight Collector is a Windows executable that communicates with the in-production SAP system. The Collector gathers performance workload and utilization statistics through the SAP NetWeaver RFC interface while having minimal impact on SAP performance. These statistics are then provided to IBM for processing and analysis. The user is very quickly provided with a detailed performance report for the workload and utilization of the SAP system.

The IBM Insight for SAP and the related documentation can be found at:

<http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS381>

This tool can be used to check the current sizing and understand the impact of incorporating users, modules, and processes, or to implement a high availability solution. The most valuable characteristic of this tool is that you are not required to fill out any form for the actual workload. This data is collected directly from the running server, on the SAP NetWeaver landscape, usually from the production system. The Insight for SAP tool only requires completing a form for the deltas and the reason for running it.

5.1.2 Compatibility

A highly available solution has multiple components and each component has to be certified and tested. This effort cannot be absorbed by the IT organization during implementation. Therefore, each provider (vendor) typically provides their own compatibility matrix, supported products and levels of each.

Verification is typically performed in a top down fashion, with the business requirements driving the starting point. For example, the latest version of SAP NetWeaver was a prerequisite in this book. The requirement for the supported LifeKeeper level was thus dependent on the SAP NetWeaver level.

The result of the sizing process helps identifying the correct level of product set required. This report has the main components for the installation, including hardware and software releases. With this high level definition, the next steps are to go deeper for each component. The high level matrix identifies a new set of hardware requisites. Subsequent analysis for eliminating single points of failure creates a requirement for additional components.

We recommend building a table with the installation components and maintain it accurately for life cycle of the installation. This not only helps the initial installation but also the test and change management procedures.

The validation process gets more complex when the versions have to be updated, for example, SAP NetWeaver SR2 has some levels of Support Packages. The combination SAP system with SPnn is supported only up to a certain SAP Kernel level. The SAP Kernel level would require some patches for specific Linux and IBM DB2. When the installation moves into production, there might be some functionality problems for which applications of some SAP Note may be required. This note might require a set of related notes or be included as a support package. This new support package requires SAP to be upgraded to a new Kernel that can create new requirements at software levels (new Linux patches or IBM DB2 version) and also at hardware levels (upgrading some Linux kernel parameters, thus requiring more hardware resources).

5.1.3 Software levels and requirements

For this book, the software levels for the main components are as follows:

- ▶ SUSE Linux Enterprise Server version 10, Service Pack 1.
- ▶ LifeKeeper 6.2.1.
- ▶ IBM DB2 9.5
- ▶ SAP NetWeaver 7.0 (previously also known as 2004s) support release 2.

SUSE Linux Enterprise Server version 10

This section illustrates the requirements for SUSE Linux Enterprise Server version 10 within a SAP NetWeaver 7.0 framework.

The following list illustrates modifications required for each kernel parameter from the default values found in `/etc/sysctl.conf`:

```
kernel.msgmni=1024
# Disable response to broadcasts.
net.ipv4.icmp_echo_ignore_broadcasts = 1
# enable route verification on all interfaces
net.ipv4.conf.all.rp_filter = 1
# enable ipv6 forwarding
#net.ipv6.conf.all.forwarding = 1
m.max_map_count = 300000
kernel.sem = 1250 256000 100 8192
kernel.shmall = 1152921504606846720
kernel.shmmax = 21474836480
```

The changes can be activated with the `sysctl -p` command.

Note: Unless otherwise stated, all commands have to be executed with super-user privileges.

LifeKeeper 6.2.1

This section illustrates the requirement for the installation of LifeKeeper 6.2.1 on a SUSE Linux Enterprise Server 10 for SAP NetWeaver.

The compatibility matrix shown in the *LifeKeeper for Linux v6 Update 2, Release Notes* document includes the combination of this LifeKeeper release on SUSE Linux Enterprise Server 10.

The complete document can be downloaded from:

<http://www.steel-eye.com/support>

LifeKeeper requires these prerequisites to be fulfilled:

- ▶ Korn Shell: At least the installation of package `pdksh-5.2.14` is required.
- ▶ Memory: The minimum memory requirement for a system supporting LifeKeeper is 128 MB on each node. This memory has to be added to the amount for memory required for running SAP NetWeaver.
- ▶ Disk space: The LifeKeeper Core Package Cluster requires approximately 61 MB for the `/opt` file system.
- ▶ Client platforms and browsers: The LifeKeeper Web client can run on any platform that provides support for Java Runtime Environment J2RE 1.4 or later. The currently supported configurations are Firefox 1.5 or 2 and Internet Explorer® 6 or 7 on Linux, Windows 2000, Windows Server® 2003, Windows XP, or Windows Vista® with J2RE 1.4, JRE™ 5, or JRE 6. Other recent platforms and browsers might work with the LifeKeeper Web client (for example, Firefox 1.0 on Linux systems, or Firefox 2 on Macintosh), but they have not been tested by SteelEye Technology®, Inc.

All hostnames and addresses have to be specified for a cluster in the client machine's local hosts file `/etc/hosts`. This minimizes the client connection time, and allows the client to connect even in the event of a Domain Name Server (DNS) failure.

- ▶ Device Mapper Multipath (DMMP): A 2.6 based Linux kernel and distribution multipath tools 0.4.5 or later.
- ▶ LifeKeeper v6.0.0 or later Core Package Cluster and requires approximately 176 Kbytes in `/opt` file system.
- ▶ Internal components for SAP NetWeaver optional recovery kit:
 - LifeKeeper v5.1.3 or later Core Package Cluster
 - LifeKeeper NFS Server Recovery Kit v5.1.0 or later
 - LifeKeeper Network Attached Storage Recovery Kit v5.0.0 or later

Based on the relational data base manager installed, the following components are also needed:

- IBM DB2 LUW Enterprise Server 9.
 - IBM DB2 Enterprise Server Edition (ESE) v8.1 or and v9.
- Oracle 10g
 - Oracle 10g Standard Edition, Standard Edition One, or Enterprise Edition
- SAP DB / MaxDB 7.5.x
 - SAP DB 7.3.0 Build 35 for use with SAP MaxDB 7.5.x

DB2 LUW Certification information

Since supported version information and details on configuration can vary due to normal product life cycle, refer to the following Web sites for current certification details:

- ▶ Certified Linux distributions for DB2:
<http://www.ibm.com/software/data/db2/linux/validate>
- ▶ DB2 hardware and software prerequisites:
<http://www.ibm.com/software/data/db2/udb/sysreqs.html>

For SAP specific information, review any current SAP Notes located on Marketplace. The following notes were available at the time of writing this book:

- ▶ SAP Note 1089578: Using DB2 9.5 with SAP software
- ▶ SAP Note 919550: DB6 SAP NetWeaver 2004s installation
- ▶ SAP Note 101809: Supported Fix Packs for DB2

Note: SAP recommends that all Fix Packs are obtained through the SAP Marketplace and not the IBM support site unless directed from the SAP support staff, because SAP software levels might differ.

DB2 LUW 9.5 Prerequisites for Linux

This section covers the requirements for DB2 LUW 9.5 Enterprise Server on a Linux operating system.

Kernel distribution requirements

DB2 running on Linux requires specific kernel parameters to ensure sufficient resources are available. These parameters have to be set on each server in the cluster. During the Linux installation, the *SAP Application Server Base* package had to be installed and many of these parameters would have been modified. These SAP modifications would appear in `/etc/sysctl.conf` with comments stating that the addition came from the SAP init function.

However, it is still necessary to verify that all the kernel parameters have been modified. Use the `ipcs` utility or `sysctl` command and analyze the output to ensure that all settings meet or exceed the minimum requirements.

The following command can be used to review the related kernel parameters:

```
/sbin/sysctl -a
```

The recommended minimum values for DB2 are shown in Table 5-1.

Table 5-1 SUSE Linux Enterprise Server kernel parameters for DB2

Kernel parameter	Recommended value (minimum)	Usage
kernel.sem	250 256000 32 1024	semaphore settings
kernel.msgmni	1024	max queues system wide
kernel.msgmax	65536	max size of message (bytes)
kernel.msgmnb	65536	default size of queue (bytes)

If the current settings are higher or equal to the minimum required values, modification is not necessary. If current values do not meet the minimum requirements, modify the settings in /etc/sysctl.conf and run sysctl with -p to load values from the parameter file.

To enable these settings to load at each boot for SUSE Linux Enterprise Server, enable boot.sysctl. The following command can be used to review the current setting for boot.sysctl:

chkconfig boot.sysctl

Package requirements

All required packages should be installed and configured before continuing with the db2 software installation. The packages shown in Table 5-2 are a prerequisite for DB2 9.5.

Table 5-2 SUSE Linux Enterprise Server packages requirement for DB2

Package name	Description
libaio	This contains the asynchronous library required for DB2.
compat-libstdc++	This contains compat-libstdc++5.0.
pdksh	This contains the korn shell, required for DPF environments.
openssh	This contains server programs for remote commands as secure shell.
rsh-server	This contains server programs for remote commands. not required if db2 is configured with ssh.

SAP NetWeaver 7.0

At the time of writing this book, these were the requisites for the SAP NetWeaver 7.0 installation:

- ▶ Korn shell: The Korn shell has to be installed in the system. To check if the Korn Shell was installed, this file has to exist:
 - /usr/bin/kshIn this case, the Korn shell was also a requisite for LifeKeeper and was already checked.
- ▶ Network File System (NFS): The NFS Server has to be enabled.
- ▶ SAP Solution Manager Key: Except in the case that the installed product was a Solution Manager, in all the other cases the Solution Manager key has to be provided.
- ▶ Java: The Java SDK has to be installed. This can be checked by following the instructions in SAP Note 709140.
- ▶ Java Cryptography Extension (JCE) policy: The JCE policy has to be installed. It can be downloaded and made available to JCE policy files from this site:
<http://www6.software.ibm.com/dl/jcesdk/jcesdk-p>
- ▶ Languages: The SAP recommendation is that all languages sets are installed or at least include the following languages:
 - Local language (including ISO8859)
 - English (including ISO8859)
 - German (including ISO8859)

It can be checked with the command:

locale -a

- ▶ File System Requirements: The file system requirements are separated based on the visibility from the servers.

The local file systems can be viewed only from each server and, in case of a failure of the server, it cannot be accessed by the other nodes of the cluster.

Table 5-3 shows the local file system with its attributes.

Table 5-3 Local file systems

File System	Size (MB)	Volume Group	Permissions	Owner:Group
/usr/sap	512	Local	775	rdbadm:sapsys
/usr/sap/RDB	3072	Local	775	rdbadm:sapsys

The shared file systems reside on a shared storage and can be accessed one at a time for each of the nodes. This is managed by a locking schema in the shared storage and its granularity is based at volume group level. The primary node takes the ownership of the volume group and establishes a lock, then until it releases that lock, any other nodes cannot have read-write access to that volume group. In the event that the lock owner fails, this lock is released and the secondary node takes the new ownership of the volume group, establishes a new lock, and gets read-write access to it.

Table 5-4 lists the attributes of the shared file systems:

Table 5-4 Shared file systems

File System	Size (MB)	Volume Group	Permissions	Owner:Group
/export/sapmnt/RDB	2048	sesapvg	775	rdbadm:sapsys
/usr/sap/RDB/DVEBMGS10	1024	sesapvg	775	rdbadm:sapsys
/usr/sap/RDB/SCS11	1024	sesapvg	775	rdbadm:sapsys
/usr/sap/RDB/ASCS10	512	sesapvg	775	rdbadm:sapsys
/export/usr/sap/trans	512	sesapvg	775	rdbadm:sapsys
/db2/db2rdb	1792	sedbvg	755	db2rdb:db2rdbadm
/db2/RDB/log_dir	1536	sedbvg	755	db2rdb:db2rdbadm
/db2/RDB/db2dump	128	sedbvg	750	db2rdb:db2rdbadm
/db2/RDB/db2rdb	256	sedbvg	750	db2rdb:db2rdbadm
/db2/RDB/saptemp1	1024	sedbvg	750	db2rdb:db2rdbadm
/db2/RDB/sapdata1	30720	sedbvg	750	db2rdb:db2rdbadm
/db2/RDB/sapdata2	10240	sedbvg	750	db2rdb:db2rdbadm
/db2/RDB/sapdata3	10240	sedbvg	750	db2rdb:db2rdbadm
/db2/RDB/sapdata4	10240	sedbvg	750	db2rdb:db2rdbadm

5.1.4 Firmware

For the test scenario in Chapter 6., “Testing and failover scenarios” on page 197, the latest version of firmware was installed for all the hardware components with special consideration to the storage components such as the Fibre Channel adapters. The recommendation is to get the hardware as current as possible, based on the compatibility matrix previously discussed. For maintenance purposes, the firmware has to be updated based on the compatibility matrix and the best practice recommendations followed, ensuring that firmware upgrades are tested in different landscapes such as development, quality assurance, pre-production and production. This is one of the reasons to get isolated hardware between the development and testing systems and the production systems.

In most cases, the SAP NetWeaver environment is not isolated from the other components and can share resources such as the Storage Area Network (SAN) with the directors or centralized storage, and the network components such as switches or middleware servers. The upgrades related to these components can also impact the SAP NetWeaver environment, or get impacted by changes on it. These have to be included for testing when some firmware has to be updated on every component. The testing scenarios have to be set up for the impact that the change can generate. For some cases, only a technical and connectivity test is adequate, and for other cases, some functional test has to be executed.

5.2 Base operating system installation

The base operating system requires certification and support by all parties delivering components to a solution. The Novell SUSE Linux Enterprise Server is supported by:

- ▶ IBM for the BladeCenter and Storage
- ▶ IBM DB2
- ▶ LifeKeeper
- ▶ SAP

The latest version of Novell SUSE Linux Enterprise Server, version 10 with the Service Pack 1 was chosen for the test scenario in this book as the latest service pack provides wide support for highly available network and storage integration.

Both servers had similar installations and were as equal as possible, with only the host names and IP addresses being the primary difference.

5.2.1 Naming

The following IP addresses were used for the test environment:

- ▶ 9.153.165.96
- ▶ 9.153.165.97
- ▶ 9.153.165.98
- ▶ 9.153.165.99

The first two were used as native host addresses, and the last two as floating addresses for the cluster installation (service IP addresses).

The netmask for all addresses was 255.255.255.0, and the default gateway was 9.153.65.1.

Because the test environment was behind a firewall, we decided to go with local addresses and host names. This avoids dependencies to other components which are probably not fulfilling high availability requirements.

Table 5-5 lists the names used internally for these IP addresses:

Table 5-5 TCP/IP address/host name mapping in the test environment

IP address	Host name
9.153.165.96	se01.redbook.lan
9.153.165.97	se02.redbook.lan
9.153.165.98	sesap.redbook.lan
9.153.165.99	sedb.redbook.lan

5.2.2 Installation

The operating system was started from a DVD media from the BladeCenter internal DVD-ROM drive, connected via an internal universal serial bus.

After the boot menu became visible, it was ended by using the **Escape** button, and an installation using a virtual network console was started. The Exiting window is shown in Figure 5-2. A virtual network console (VNC) installation was chosen to allow access to the BladeCenter console available to other users of the BladeCenter quickly.

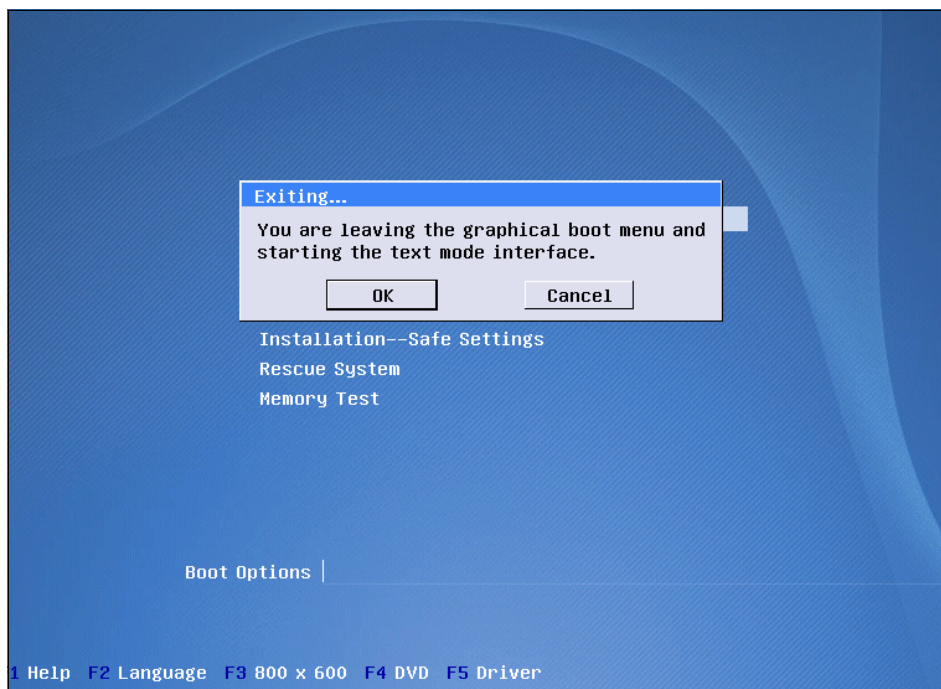


Figure 5-2 Operating system installation; graphical boot menu

At the boot prompt, the parameters for a network configuration and the virtual network console installation had to be specified.

These were:

- ▶ `vnc=1` is used to enable the virtual network console.
- ▶ `vncpassword=<insert_password>` to set a password for the installation. This is mandatory; it has to be at least eight characters long and requires at least a little complexity.
- ▶ `hostip=9.153.165.96/24` specifies the IP address of the server and the netmask to be used.
- ▶ `gateway=9.153.165.1` specifies a gateway for the default route, which is required because the administrators' desktop personal computers are in a different network segment.

The sample VNC command is shown in Example 5-1.

Example 5-1 VNC command

```
boot: linux vnc=1 vncpassword=red8book hostip=9.153.165.96/24  
gateway=9.153.165.1
```

This brings up the installation environment with a YaST installer on a virtual desktop, accessible through a virtual network console client.

On a client PC, a virtual network console client was started to continue the installation by issuing the following command:

vncviewer 9.153.165.96:1

The virtual network console was running on display 1 (or port the 5901, TCP). If there was no console client software available, it is also possible to use applet based client through a Web browser by directing to <http://9.153.65.96:5801>.

In the following figures we cover those steps of the installation that required special attention regarding the planned scenario.

In the SAP installation documentation, there are specific hints for the Linux base operating system installation that must be followed.

This information can be obtained in SAP Note 958253 from the SAP Marketplace by searching for it from this link:

<http://service.sap.com/notes>

Note: The SAP Marketplace is a secure site and requires having a registered user and password.

Language

The language for both the installation and the running system is expected to be English, according to the SAP installation documentation. Figure 5-3 shows the language selection in the installation process.

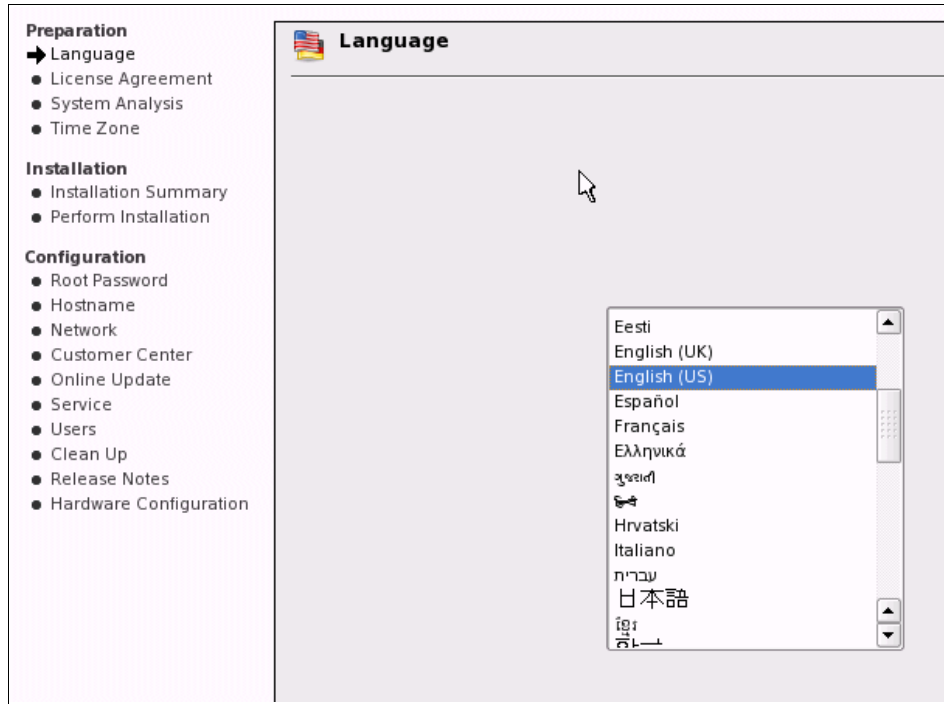


Figure 5-3 Language selection

Time zone setting

The system clock was configured to universal time and the proper time zone for Germany, which also covers daylight savings time. Figure 5-4 shows the window with the time zone setting.

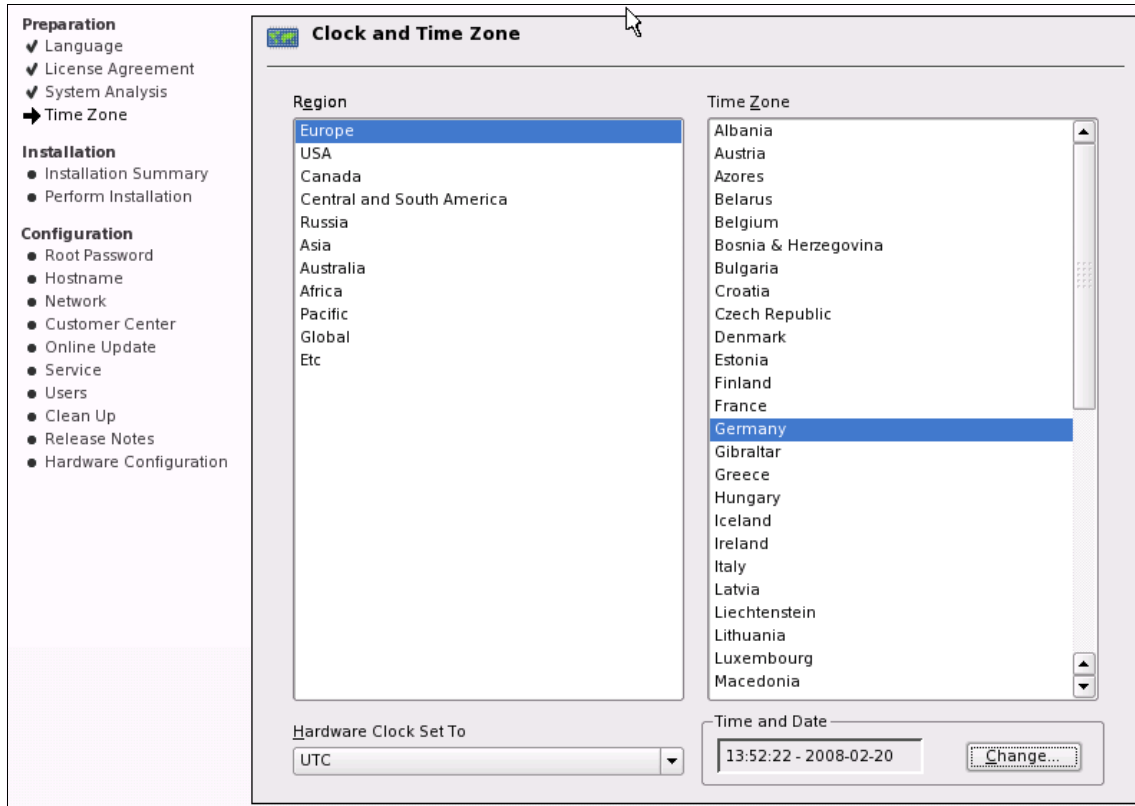


Figure 5-4 Time zone setting

Internal disks

The disk partition was created different from the standard. The partition layout was created with several aspects in mind:

- ▶ Maximum protection against data loss
- ▶ Best usage of the file system
- ▶ Independency between local and cluster file system
- ▶ Easy enhancement

This means that the file systems have been split up so that the risk of losing data of open files becomes lower when there is less traffic on a file system. The file systems have all been created with as much capacity as required, but no more. So it was required to be able to resize it easily, even while it is in use. All this is provided by the XFS file system, so that was chosen for the internal disks.

Only the /boot partition is a native Linux partition containing the boot loader. Everything else goes into one big partition that is used as a physical volume for a system group containing all the operating system's file systems.

One major SAP requirement is that the virtual memory, the swap space size of the base operating system, has to be created twice as big as the physical memory of the server. The test machines have been equipped with 8 GByte of memory, so the swap space was created with 16 GByte.

Partitioning

Because the partitioning proposals from the installer did not fit the requirements of the tests, a custom partition setup was created, as shown in Figure 5-5.

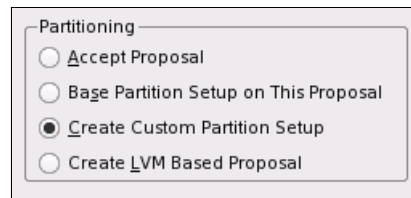


Figure 5-5 Create custom partition setup

The servers were equipped with two internal disks. For a high availability scenario, they should be mirrored to each other (RAID 1) to prevent suffering from disk failure. For the test environment, it was chosen to go with single disks, as we show in Figure 5-6. As these disks do not go into cluster management at all, the cluster setup does not change.

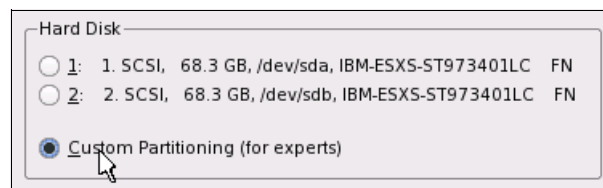


Figure 5-6 Custom partitioning

After erasing all existing partitions, two partitions were created, as shown in Figure 5-7, in the first internal disk:

- ▶ The first partition /dev/sda1 contains the Linux boot loader GRUB.
- ▶ The second partition /dev/sda2 is a physical volume for a volume group containing the base operating system.

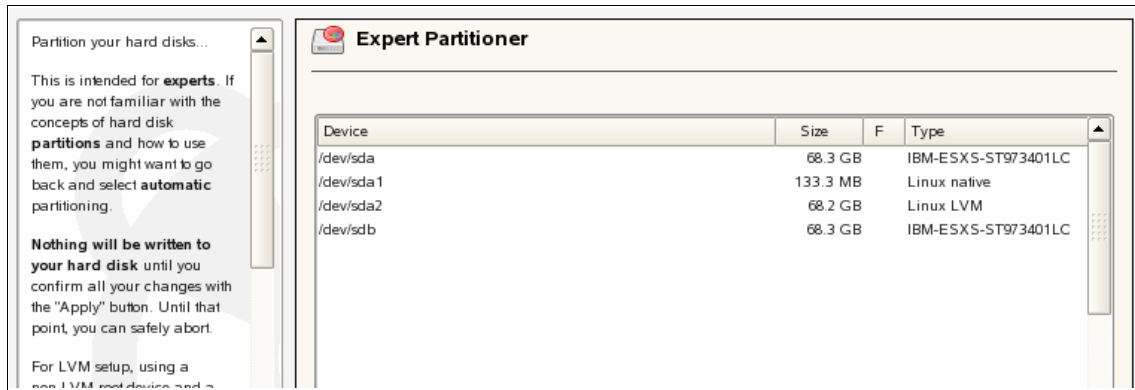


Figure 5-7 Disk partitioning

Example 5-2 shows the disk partitioning from the command line.

Example 5-2 disk output

```
se01:/etc # fdisk -l /dev/sda
```

```
Disk /dev/sda: 73.4 GB, 73407488000 bytes
255 heads, 63 sectors/track, 8924 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	17	136521	83	Linux
/dev/sda2		18	8924	71545477+	8e	Linux LVM

```
se01:/etc #
```

Logical volume management

In the logical volume management of the SUSE Linux Enterprise Server installer, a volume group called `systemvg` was created. The second partition of the first disk, `/dev/sda2`, was added as a physical volume to that partition, as shown in Figure 5-8.

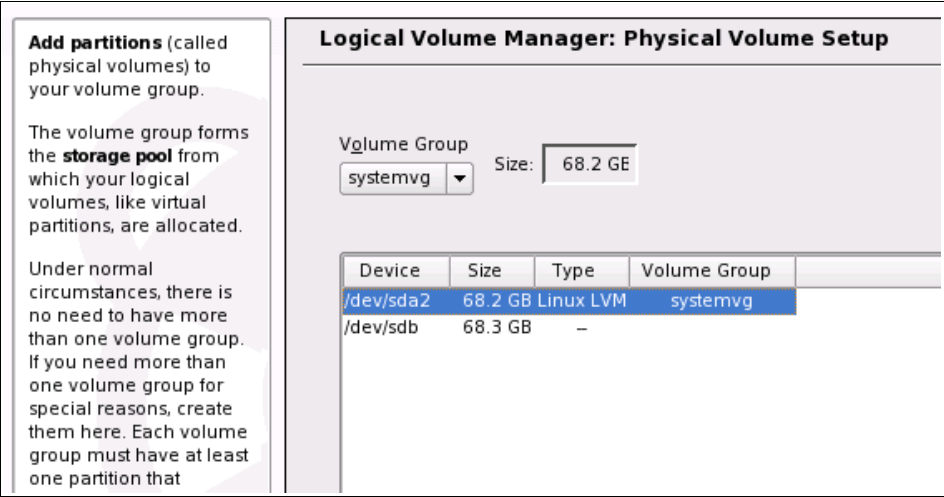


Figure 5-8 Create a volume group

Example 5-3 shows the same data viewed with command line tools.

Example 5-3 Physical volumes

```
se01:/etc # pvs
  PV          VG          Fmt  Attr PSize   PFree
/dev/sda2    systemvg    lvm2  a-    68.23G  41.48G
se01:/etc #
```

After creating the volume group, a couple of logical volumes have been created. The naming rule for the base operating system was to use the mount point name followed by `lv`. The root file system goes to `rootlv`, the `/var` file system to `varlv`, and so on.

Figure 5-9 provides an overview of the overall disk layout during the base operating system installation.

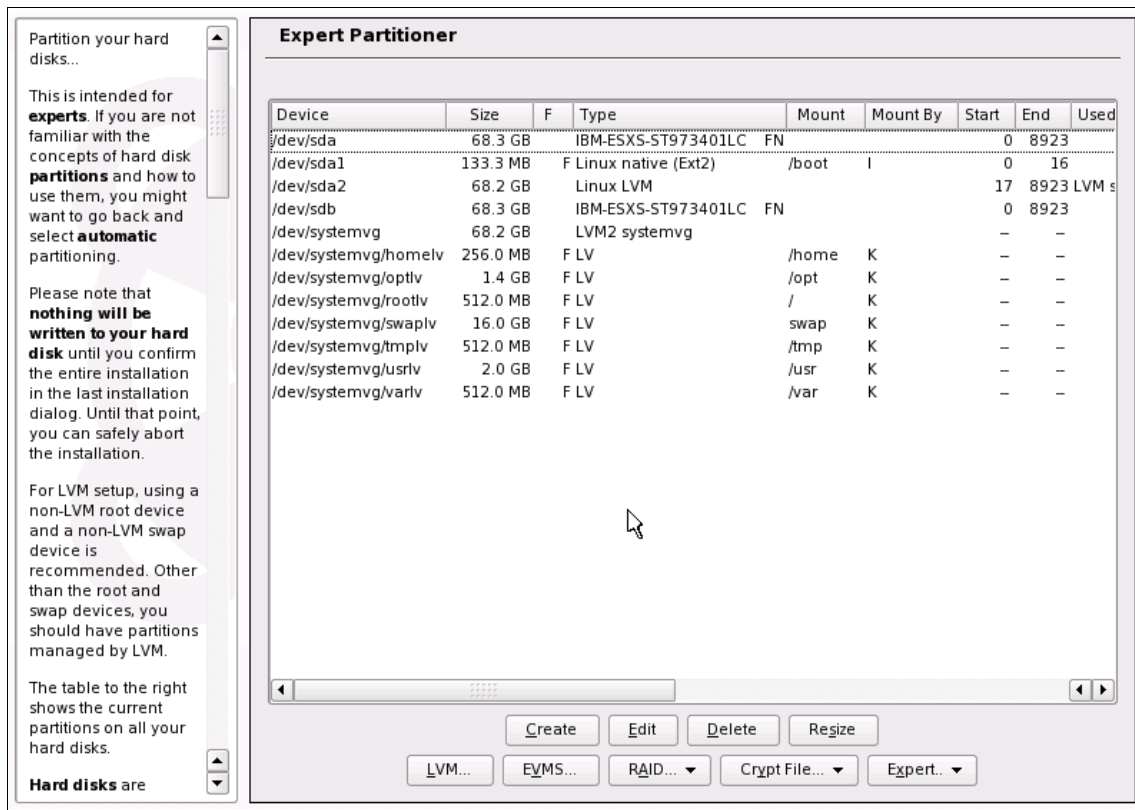


Figure 5-9 Overall disk layout

After the installation, the list of logical volumes on that server are shown on the command line in Example 5-4.

Example 5-4 Logical volumes

```
se01:~ # lvs
LV          VG      Attr  LSize   Origin Snap%  Move Log Copy%
homelv      systemvg -wi-ao 256.00M
optlv       systemvg -wi-ao  1.50G
rootlv      systemvg -wi-ao 512.00M
swaplv      systemvg -wi-ao 16.00G
tmplv       systemvg -wi-ao 512.00M
usrlv       systemvg -wi-ao  2.00G
varlv       systemvg -wi-ao 512.00M
se01:~ #
```

Software installation

The installation was based on the defaults. Following the SAP installation requirements, the *SAP Application Server Base* and the *C/C++ Compiler and Tools* were added to the installation. The *Novell AppArmor®* features were disabled. The configuration window is shown in Figure 5-10.

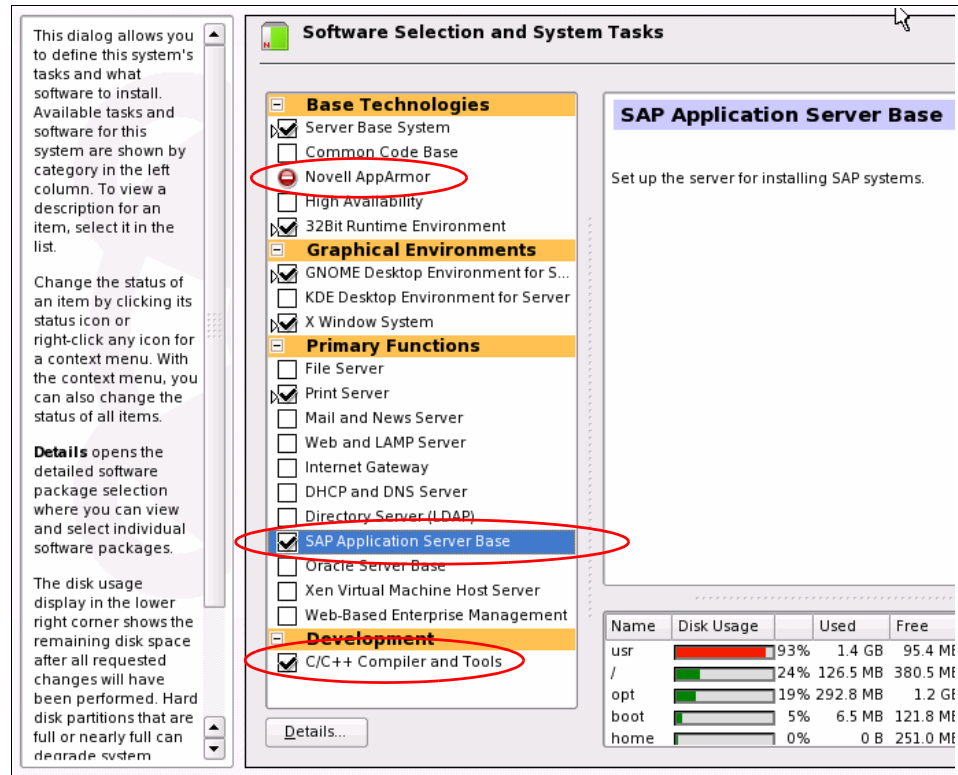


Figure 5-10 Software selection

Confirming the installation

After choosing all required options, the installation was started. Figure 5-11 shows the installation summary.

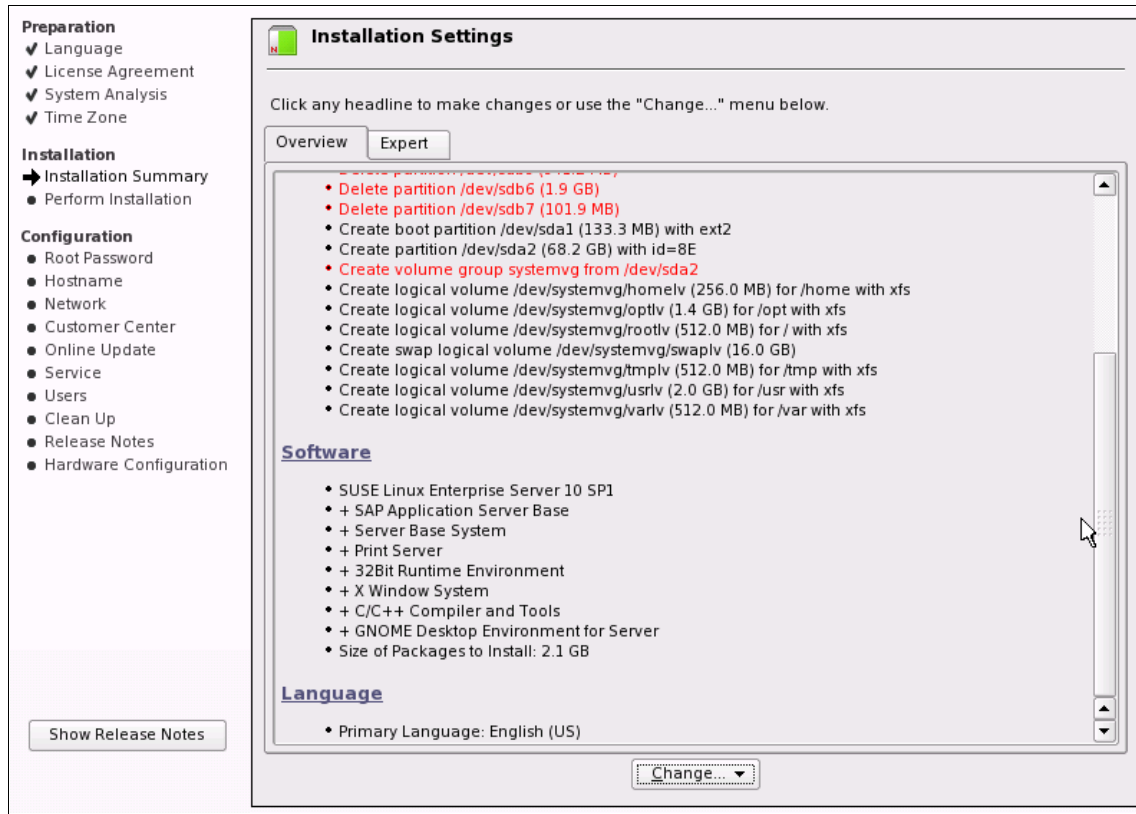


Figure 5-11 Installation settings

Further configuration during installation

After creating file systems and copying all files to the internal disks, the server rebooted from there. During the post-installation tasks, the following steps were performed:

1. Disable SUSEfirewall2.
2. Disable CUPS listening to remote printers.
3. Allow remote administration via SSH.
4. Skip generating SSL certificates.
5. Create a root user, but no further users.
6. Do network configuration.

Network configuration

The network configuration for both internal network adapters was performed during the installation process. The traditional method using the `ifup` command was chosen as shown in Figure 5-12.

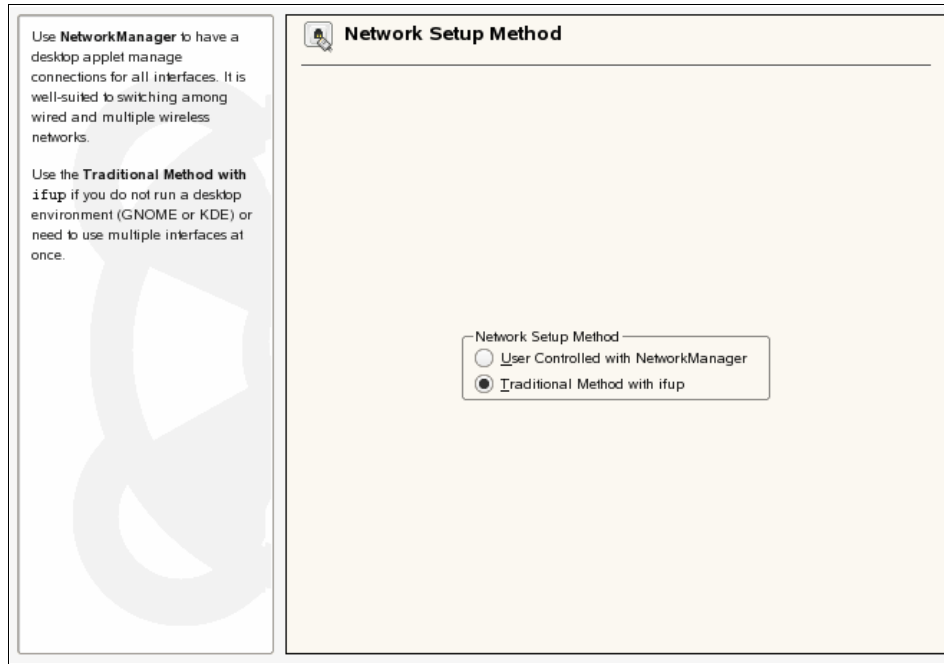


Figure 5-12 Network setup method

The network adapters were recognized automatically by the SUSE Linux Enterprise Server operating system. The first adapter was configured for external communication, while the second one was configured as a heartbeat adapter for internal communication only, as shown in Figure 5-13.

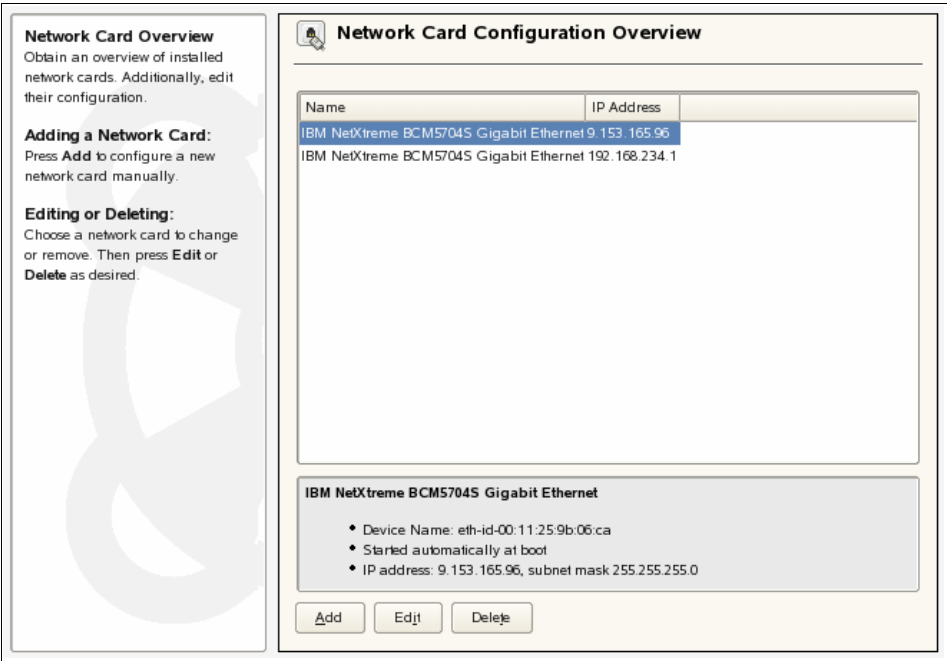


Figure 5-13 Network card configuration overview

The server addresses were configured as static addresses, as shown in Figure 5-14.

The screenshot displays the 'Network Address Setup' window with two panes. The left pane contains instructional text about address setup options. The right pane, titled 'Network Address Setup', has two tabs: 'General' and 'Address'. The 'Address' tab is active, showing configuration fields for 'Device Type' (set to 'Ethernet') and 'Configuration Name' (set to 'id-00:11:25:9b:06:ca'). Three radio buttons are present: 'None Address Setup', 'Automatic Address Setup (via DHCP)', and 'Static Address Setup' (which is selected). Below these, the 'IP Address' field contains '9.153.165.96' and the 'Subnet Mask' field contains '255.255.255.0'. A 'Detailed Settings' section at the bottom includes buttons for 'Hostname and Name Server', 'Routing', and an 'Advanced...' dropdown menu.

You can select none address setup if you don't want any IP address.

You can select dynamic address assignment if you have a **DHCP server** running on your local network.

Also select this if you do not have a static IP address assigned by the system administrator or your cable or DSL provider.

Network addresses are then obtained **automatically** from the server.

Otherwise, network addresses must be assigned **manually**.

Enter the IP address (e.g., 192.168.100.99) for your computer, the network mask (usually 255.255.255.0), and, optionally, the default gateway IP address.

Clicking **Next** completes the configuration.

Network Address Setup

General Address

Device Type: Ethernet Configuration Name: id-00:11:25:9b:06:ca

☐ None Address Setup
☐ Automatic Address Setup (via DHCP)
☒ Static Address Setup

IP Address: 9.153.165.96

Subnet Mask: 255.255.255.0

Detailed Settings

Hostname and Name Server

Routing

Advanced...

Figure 5-14 Network address setup (address)

The host name and domain name have been set statically, and no name servers were used, as shown in Figure 5-15.

Enter the name for this computer and the DNS domain that it belongs to.

Optionally enter the name server list and domain search list.

Note that the hostname is global—it applies to all interfaces, not just this one.

The domain is especially important if this computer is a mail server.

If you are using DHCP to get an IP address, check whether to get a hostname via DHCP. The hostname of your host (which can be seen by issuing `hostname` command) will be set automatically by DHCP client. You may want to disable this option if you connect to different networks that might each assign a different hostname, because changing the hostname at runtime may confuse the graphical desktop.

If you are using DHCP to get an

Hostname and Name Server Configuration

Hostname and Domain Name (Global)

Hostname: Domain Name:

☐ Change Hostname via DHCP

☒ Write Hostname to /etc/hosts

Name Servers and Domain Search List

Name Server 1:

Name Server 2:

Name Server 3:

Domain Search:

☒ Update Name Servers and Search List via DHCP

Figure 5-15 Hostname and name server configuration

The default gateway was configured as the gateway as shown in Figure 5-16.

The routing can be set up in this dialog. The **Default Gateway** matches every possible destination, but poorly. If any other entry exists that matches the required address, it is used instead of the default route. The idea of the default route is simply to enable you to say "and everything else should go here."

Enable **IP Forwarding** if the system is a router.

Routing Configuration

Default Gateway
9.153.165.1

Routing Table

☐ Expert Configuration

Destination	Gateway	Netmask	Device	Options
-------------	---------	---------	--------	---------

Add Edit Delete

☐ Enable IP Forwarding

Figure 5-16 Routing configuration

The firewall was disabled, and the start of the interface at boot time was enabled as shown in Figure 5-17.

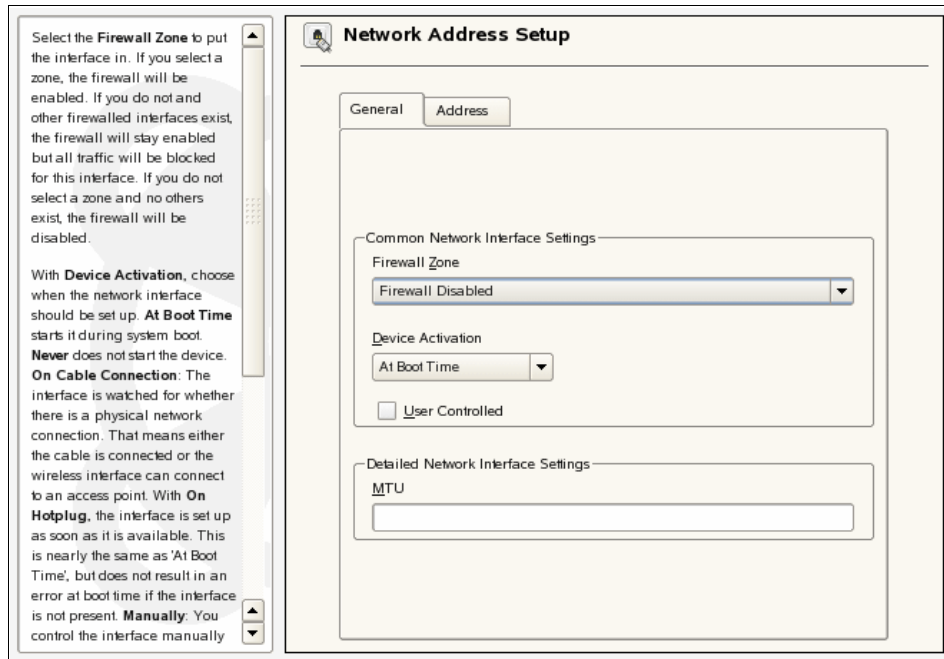


Figure 5-17 Network address setup (general)

Post-installation tasks

A couple of further post-installation tasks were performed manually.

Disabling unnecessary services

The following services were disabled, because they were not required for the test scenario:

- ▶ alsasound
- ▶ cups
- ▶ cupsrenice
- ▶ novell-zmd
- ▶ nscd
- ▶ powersaved

Windows manager configuration

The Graphical User Interface was configured so that it did not start a GUI on the console, but instead allowed connection through a virtual network console as shown in Example 5-5.

```
## Type:      string
## Default:
## Path: Desktop
## Description: default mouse cursor theme
...
...
...
## Type:      yesno
## Default:    no
#
# Allow remote access (XDMCP) to your display manager (xdm/kdm/gdm).
Please note
# that a modified kdm or xdm configuration, e.g. by KDE control center
# will not be changed. For gdm, values will be updated after change.
# XDMCP service should run only on trusted networks and you have to
disable
# firewall for interfaces, where you want to provide this service.
#
DISPLAYMANAGER_REMOTE_ACCESS="yes"

## Type:      yesno
## Default:    no
#
# Allow remote access of the user root to your display manager. Note
# that root can never login if DISPLAYMANAGER_SHUTDOWN is "auto" and
# System/Security/Permissions/PERMISSION_SECURITY is "paranoid"
#
DISPLAYMANAGER_ROOT_LOGIN_REMOTE="yes"

## Type:      yesno
## Default:    yes
#
# Let the displaymanager start a local Xserver.
# Set to "no" for remote-access only.
# Set to "no" on architectures without any Xserver (e.g. s390/s390x).
#
DISPLAYMANAGER_STARTS_XSERVER="no"

## Type:      yesno
## Default:    no
#
# TCP port 6000 of Xserver. When set to "no" (default) Xserver is
# started with "-nolisten tcp". Only set this to "yes" if you really
```

```
# need to. Remote X service should run only on trusted networks and
# you have to disable firewall for interfaces, where you want to
# provide this service. Use ssh X11 port forwarding whenever possible.
#
DISPLAYMANAGER_XSERVER_TCP_PORT_6000_OPEN="no"
...
...
...
```

Daemon to execute scheduled command

The cron daemon was modified, without disturbing the daily operation of the servers, because it performs daily maintenance tasks during the night, as shown in Example 5-6.

Example 5-6 Modifying /etc/sysconfig/cron

```
## Path:      System/Cron
## Description: days to keep old files in tmp-dirs, 0 to disable
## Type:      integer
## Default:   0
## Config:
...
...
...
# Type:      time (eg: 14:00)
# Default:   nothing
#
# At which time cron.daily should start. Default is 15 minutes after
booting
# the system. Due the cron script runs only every 15 minutes, it will
only
# run on xx:00, xx:15, xx:30, xx:45, not at the accurate time you set.
DAILY_TIME="03:00"
...
...
...
```

Network time protocol

SAP recommends having a time server to synchronize machines running an SAP infrastructure. On the test environment, there was no network time protocol server available, so one of the servers was set up as a time server and the other synchronized from it. Even if the time was not correct, it was synchronous on both servers.

The first server se01 was configured to act a time server, the second se02 was configured to be its client, as shown in Figure 5-18.

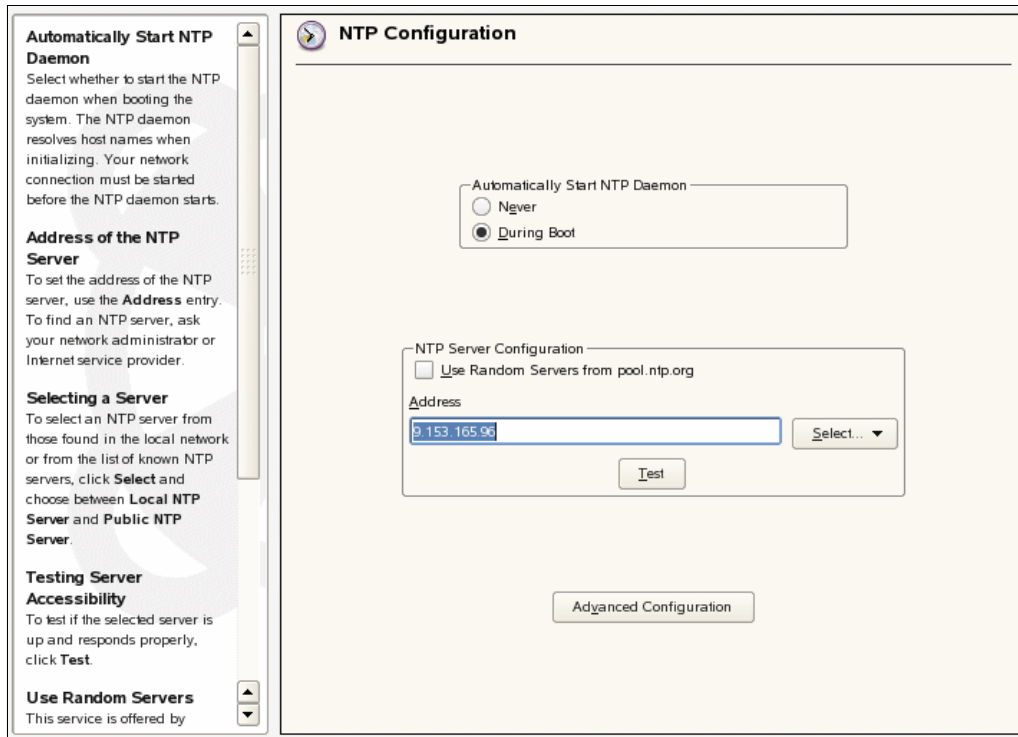


Figure 5-18 Network time protocol configuration

Enable SSH for cluster

The DB2 database software uses a remote shell (rsh) connection to start a database instance if the instance is configured to allow connections through a service IP address that is different from the host native's address. The hosts have to be configured in the .rhosts file of a user or in the system wide /etc/hosts.equiv.

Together with rexec, rlogin, and rcp (remote execution, remote copy), rsh is a very old protocol and does not offer much security. Its authentication is based on clear text passwords or TCP/IP addresses, and all input and output is unencrypted. This makes it vulnerable for spoofing attacks, and even in a firewall protected switched network segment, it cannot be seen as secure.

OpenSSH is a secure shell implementation that comes with most Linux distributions. It allows remote login, remote command execution, and remote copying through. All communication is encrypted. Secure shell offers authentication based on public-key.

OpenSSH creates public keys for a host it is running on. As soon as a user logs into a server, the server's public key is stored on the client machine. For Linux, this is the file `~.ssh/known_hosts`. The next time the user logs in to a server, the host key is checked and the user is informed if the server's key has changed. This protects from TCP/IP address spoofing, but leads to difficulties in a cluster environment.

A DB2 database under control of LifeKeeper uses the `ssh` command. This just connects to the host it is running on, but it requires at least a valid private key/public key combination without password protection on that node and the public key to be allowed to log in.

In the test scenario for easier maintenance, it was decided to enable host based authentication. That means, a user that is has been authenticated successfully by one node of the cluster is able to login with repeated authentication to the other node.

This requires the host based authentication to be enabled on both the server and the client, and it requires the host public keys to be put into a special file.

Example 5-7 shows the changes that have been performed to the configuration file `/etc/ssh/sshd_config` on both nodes `se01` and `se02`. The Secure Shell Daemon needs to be restarted after the change, for example, using command `/etc/init.d/sshd restart`.

Example 5-7 Changes in `/etc/ssh/sshd_config`

```
...
RSAAuthentication yes
#PubkeyAuthentication yes
#AuthorizedKeysFile      .ssh/authorized_keys

# For this to work you will also need host keys in
/etc/ssh/ssh_known_hosts
RhostsRSAAuthentication yes
# similar for protocol version 2
HostbasedAuthentication yes
# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes
...
```

In the client configuration file, `/etc/ssh/ssh_config`, the following changes have been made. Example 5-8 shows the modified settings.

Example 5-8 Changes in `/etc/ssh/ssh_config`

```
...
ForwardX11Trusted yes
RhostsRSAAuthentication yes
#   RSAAuthentication yes
#   PasswordAuthentication yes
    HostbasedAuthentication yes
...
EnableSSHKeysign yes
...
```

To enable a user with a host's private key to authenticate a user's key, the `ssh` client command needs root privileges. All private keys can either be set globally for the `ssh` command, or just for the `ssh-keysign` component. Since OpenSSH 3.6, it is no longer set as the default; this configuration has to be performed manually. Execute the following command on both of the nodes in the cluster:

```
chmod u+s /usr/lib64/ssh/ssh-keysign
```

The public keys of both the nodes must be put in the `/etc/ssh/ssh_known_hosts` file on both nodes. Because the file did not exist, it was created as an empty file using the `touch` command, and the keys were added using the `ssh-keyscan` command. Both nodes have to be in that file to allow `ssh` functionality for each local node to the same node and for each local node to a remote node.

The `ssh-keyscan` command contains the key type, node names, and aliases that have to be put into the file:

- ▶ **ssh-keyscan -t rsa se01,se01.redbook.1an,9.153.165.96**
- ▶ **ssh-keyscan -t rsa se02,se02.redbook.1an,9.153.165.97**

Example 5-9 shows creation of the `/etc/ssh/ssh_known_hosts` configuration file.

Example 5-9 Example of creating the `/etc/ssh/ssh_known_hosts` file

```
se01:/etc/ssh # touch ssh_known_hosts
se01:/etc/ssh # ssh-keyscan -t rsa se01,se01.redbook.1an,9.153.165.96 >
ssh_known_hosts
# se01 SSH-1.99-OpenSSH_4.2
se01:/etc/ssh # ssh-keyscan -t rsa se02,se02.redbook.1an,9.153.165.97 >
ssh_known_hosts
# se02 SSH-1.99-OpenSSH_4.2
se01:/tmp/ssh # cat ssh_known_hosts
```

```
se02,se02.redbook.lan,9.153.165.97 ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAzREb0w/Mua9dn1EJquz7ziZjC51exwRF7mBrio2N94a
k4n8Ebro8ZjmVjt2TKTQ4uZARG5xNDiIrNjD859af2SN4hDb5CmdRqVVH4E4B2ddQSx20qG
bM2qkucced0tMGBc79e0hqq+Ne9qaQ6o25g/1eW0LCG1RgsaVqSrMjH9U=
se01:/etc/ssh #
```

Example 5-10 shows the result of the former action, the content of the configuration file /etc/ssh/ssh_known_hosts:

Example 5-10 Content of /etc/ssh/ssh_known_hosts

```
se01,se01.redbook.lan,9.153.165.96 ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAuMDuu0Ieby/5Q1oEA/XtrW3tRWITjnpngCQH3FFNSf
2t5kVjTsAdDASzYIUC8/xwYTiekVk8QHgnw6j/IT4djYAT0vdaCD61Ps27VN9HT3cmFSEaC
PsS2ZZe7sLB14Jb0cfEAmp8SqlpvZ+hCSKfxtCA1sdj/aXLAN8exg+0
3U=
se02,se02.redbook.lan,9.153.165.97 ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAzREb0w/Mua9dn1EJquz7ziZjC51exwRF7mBrio2N94a
k4n8Ebro8ZjmVjt2TKTQ4uZARG5xNDiIrNjD859af2SN4hDb5CmdRqVVH4E4B2ddQSx20qG
bM2qkucced0tMGBc79e0hqq+Ne9qaQ6o25g/1eW0LCG1RgsaVqSrMjH
9U=
```

5.3 Shared storage

This section discusses the configuration of the operating system for accessing the shared storage residing on the Storage Area Network.

5.3.1 Host bus adapters

The on-board Qlogic host bus adapters are recognized automatically and the required driver qla2xxx is loaded during system startup. That driver recognizes all SCSI disks it sees and gives them names following the schema for Linux SCSI disk naming (/dev/sda, /dev/sdb and so on). The driver for the internal SCSI disks, mptscsih, is loaded before the Qlogic Fibre Channel driver.

For this environment, the internal SCSI controller for the internal disks in the Blade Server gets bus number 0, while the bus numbers for the two individual Qlogic controllers become 1 and 2.

Because the hardware drivers cannot distinguish if they see the same device through multiple connections, each appearance will be recognized as an individual disk, as shown in Example 5-11.

Example 5-11 Overview about SCSI busses and devices

```
se01:~ # ls SCSI
[0:0:0:0] disk IBM-ESXS ST973401LC FN B418 /dev/sda
[0:0:1:0] disk IBM-ESXS ST973401LC FN B41D /dev/sdb
[1:0:0:0] no dev IBM 2145 0000 -
[1:0:0:11] disk IBM 2145 0000 /dev/sdc
[1:0:0:12] disk IBM 2145 0000 /dev/sdd
[1:0:0:13] disk IBM 2145 0000 /dev/sde
[1:0:0:14] disk IBM 2145 0000 /dev/sdf
[1:0:1:0] no dev IBM 2145 0000 -
[1:0:1:11] disk IBM 2145 0000 /dev/sdg
[1:0:1:12] disk IBM 2145 0000 /dev/sdh
[1:0:1:13] disk IBM 2145 0000 /dev/sdi
[1:0:1:14] disk IBM 2145 0000 /dev/sdj
[2:0:0:0] no dev IBM 2145 0000 -
[2:0:0:11] disk IBM 2145 0000 /dev/sdk
[2:0:0:12] disk IBM 2145 0000 /dev/sdl
[2:0:0:13] disk IBM 2145 0000 /dev/sdm
[2:0:0:14] disk IBM 2145 0000 /dev/sdn
[2:0:1:0] no dev IBM 2145 0000 -
[2:0:1:11] disk IBM 2145 0000 /dev/sdo
[2:0:1:12] disk IBM 2145 0000 /dev/sdp
[2:0:1:13] disk IBM 2145 0000 /dev/sdq
[2:0:1:14] disk IBM 2145 0000 /dev/sdr
se01:~ #
```

With the storage area network topology and the configuration in mind, it can be noted that there is a group of four logical unit numbers (11, 12, 13, 14) that was discovered four times. From this output, it cannot be recognized which devices are unique and which point to the same logical unit number. The most reliable way of determining the Storage Area Network volume to which the Linux SCSI disk name points, is to verify the world wide name. Every volume has a logical unit number and a world wide name.

Example 5-12 gathers the world wide names for each Linux SCSI disk device and creates a list of world wide names and the disk devices pointing to it.

Example 5-12 Example script showing devices grouped by WWN, show_equal_wwn

```
#!/bin/sh

DISKLISTFILE=/tmp/$$
cd /dev
for disk in sd[a-z] sd[a-z][a-z]
do
```

```

        echo -n "${disk} "
        scsi_id -g -u -s /block/${disk}
done > $DISKLISTFILE

WWNLIST=$( cat $DISKLISTFILE | cut -d " " -f 2 | sort | uniq )

echo ""
echo "SCSI devices sorted by WWN"
echo "=====

for WWN in $WWNLIST
do
    echo "WWN: $WWN"
    grep $WWN $DISKLISTFILE | awk '{ print "    /dev/"$1 }'
done

```

Example 5-13 shows a list of all Linux SCSI disk devices recognized by the operating system, grouped by their world wide name. This list shows, for example, that the Linux SCSI disk devices sdc, sdg, sdk and sdo point to exactly the same world wide name, and that means to the same logical unit number on the storage area network. Behind these different devices, these are four different routes to the storage volume and are four different paths.

Example 5-13 List of Linux SCSI disk devices grouped by WWN

```

se01:~ # ./show_equal_wnn
SCSI devices sorted by WWN
=====
WWN: 3600507680185000d3800000000000053
    /dev/sdc
    /dev/sdg
    /dev/sdk
    /dev/sdo
WWN: 3600507680185000d3800000000000054
    /dev/sdd
    /dev/sdh
    /dev/sdl
    /dev/sdp
WWN: 3600507680185000d3800000000000055
    /dev/sde
    /dev/sdi
    /dev/sdm
    /dev/sdq
WWN: 3600507680185000d3800000000000056
    /dev/sdf

```

```
/dev/sdj
/dev/sdn
/dev/sdr
WWN: SIBM-ESXSST973401LC_F3LB05EFA00007603ALBD
/dev/sda
WWN: SIBM-ESXSST973401LC_F3LB0EEGN00007631J2LX
/dev/sdb

se01: ~#
```

5.3.2 Configuring multipath connectivity

The multipath module configuration resides in the file `/etc/multipath.conf`. This file was modified to recognize the IBM SAN Volume Controller with the correct personality module, and alias names for the disks to be used were created. The services `boot.multipath` and `multipathd` were enabled for system boot through the **chkconfig** command and the server was rebooted. This change is shown in Example 5-14.

Example 5-14 Linux multipath configuration file, `/etc/multipath.conf`

```
devnode_blacklist {
    ## Replace the wwid with the output of the command
    ## 'scsi_id -g -u -s /block/[internal scsi disk name]'
    ## Enumerate the wwid for all internal scsi disks.

    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st|sda|sdb)[0-9]*"
    devnode "^hd[a-z] [[0-9]*]"
}

defaults {
    polling_interval 30
}

devices {
    # Recognizing an IBM SAN Volume Controller
    device {
        vendor            "IBM"
        product           "2145"
        path_grouping_policy group_by_prio
        prio_callout      "/sbin/mpath_prio_alua /dev/%n"
    }
}
```

```

multipaths {
    multipath {
        wwid      3600507680185000d38000000000000053
        alias     sedbd0
    }
    multipath {
        wwid      3600507680185000d38000000000000054
        alias     sedbd1
    }
    multipath {
        wwid      3600507680185000d38000000000000055
        alias     sesapd0
    }
    multipath {
        wwid      3600507680185000d38000000000000056
        alias     sesapd1
    }
}

```

After rebooting, the server recognizes the shared drives and creates aliases for accessing the multipath devices, shown by the **multipath -l** command in Example 5-15.

Example 5-15 Output of the multipath command

```

sedbd1 (3600507680185000d38000000000000054) dm-12 IBM,2145
[size=128G][features=0][hw_handler=0]
\_ round-robin 0 [prio=0][active]
\_ 1:0:0:12 sdd 8:48 [active][undef]
\_ 2:0:0:12 sdl 8:176 [active][undef]
\_ round-robin 0 [prio=0][enabled]
\_ 1:0:1:12 sdh 8:112 [active][undef]
\_ 2:0:1:12 sdp 8:240 [active][undef]
sedbd0 (3600507680185000d38000000000000053) dm-11 IBM,2145
[size=128G][features=0][hw_handler=0]
\_ round-robin 0 [prio=0][active]
\_ 1:0:1:11 sdg 8:96 [active][undef]
\_ 2:0:1:11 sdo 8:224 [active][undef]
\_ round-robin 0 [prio=0][enabled]
\_ 1:0:0:11 sdc 8:32 [active][undef]
\_ 2:0:0:11 sdk 8:160 [active][undef]
sesapd1 (3600507680185000d38000000000000056) dm-17 IBM,2145
[size=64G][features=0][hw_handler=0]
\_ round-robin 0 [prio=0][active]
\_ 1:0:0:14 sdf 8:80 [active][undef]

```



```

\_ 2:0:0:14 sdn 8:208 [active][undef]
\_ round-robin 0 [prio=0][enabled]
\_ 1:0:1:14 sdj 8:144 [active][undef]
\_ 2:0:1:14 sdr 65:16 [active][undef]
SIBM-ESXSST973401LC_F3LB0EEGN00007631J2LXd-10 IBM-ESXS,ST973401LC
[size=68G][features=0][hwhandler=0]
\_ round-robin 0 [prio=0][active]
\_ 0:0:1:0 sdb 8:16 [active][undef]
sesapd0 (3600507680185000d3800000000000055) dm-13 IBM,2145
[size=64G][features=0][hwhandler=0]
\_ round-robin 0 [prio=0][active]
\_ 1:0:1:13 sdi 8:128 [active][undef]
\_ 2:0:1:13 sdq 65:0 [active][undef]
\_ round-robin 0 [prio=0][enabled]
\_ 1:0:0:13 sde 8:64 [active][undef]
\_ 2:0:0:13 sdm 8:192 [active][undef]

```

Further documentation on SAN Volume Controller is available in the *IBM System Storage SAN Volume Controller*, SG24-6423, and at the following Web site:

http://www-1.ibm.com/support/docview.wss?rs=591&uid=s591S1002865#_Red_Hat_Linux

5.3.3 Configuring host-based mirroring

Host-based mirroring was set up using by creating a Software RAID mirror through the command line interface as shown in Example 5-16.

Example 5-16 Creating Software RAID mirrors

```

mdadm --create /dev/md0 --raid-devices=2 --level=1 \
      /dev/disk/by-name/sedbd0 /dev/disk/by-name/sedbd1

mdadm --create /dev/md1 --raid-devices=2 --level=1 \
      /dev/disk/by-name/sesapd0 /dev/disk/by-name/sesapd1

```

Note: The back slash (\) used in this example at the end of each line marks the continuation of the next line.

To be able to start and stop one or both of these devices on both servers se01 and se02, the configuration was put into the configuration file `/etc/mdadm.conf`. The UUID of the RAID devices was chosen to make it work independent from any changes in the device naming. The UUID can be found using the `mdadm` command as shown in Example 5-17.

Example 5-17 Output of mdadm --query --detail /dev/md1

```
Version : 00.90.03
  Creation Time : Thu Feb 21 15:53:27 2008
    Raid Level : raid1
    Array Size : 134214912 (128.00 GiB 137.44 GB)
  Used Dev Size : 134214912 (128.00 GiB 137.44 GB)
    Raid Devices : 2
    Total Devices : 2
Preferred Minor : 1
  Persistence : Superblock is persistent

    Update Time : Wed Feb 27 12:13:39 2008
      State : clean
Active Devices : 2
Working Devices : 2
Failed Devices : 0
Spare Devices : 0

      UUID : e996777e:484327ff:cb25cebc:77e7d0df
    Events : 0.188969
```

Number	Major	Minor	RaidDevice	State	
0	253	17	0	active sync	/dev/dm-17
1	253	15	1	active sync	/dev/dm-15

This UUID and the corresponding device name has been put into */etc/mdadm.conf* as shown in Example 5-18. This file was copied to the other server after finishing it.

Example 5-18 Configuration file /etc/mdadm.conf

```
ARRAY /dev/md0
    level=raid1
    num-devices=2
    UUID=eecd5458:a66a1f98:4eeda444:9de774a3

ARRAY /dev/md1
    level=raid1
    num-devices=2
    UUID=e996777e:484327ff:cb25cebc:77e7d0df

MAILADDR root@localhost
```

The mdadm (multiple devices administrator) driver must not assemble any array automatically during system startup. The appropriate parameter was set in the file `/etc/sysconfig/mdadm`, is set up, and is shown in Example 5-19.

Example 5-19 Mdadm configuration

```
## Type: yesno
## Default: no
#
# "yes" for mdadm.conf to be used for array assembly on boot
#
BOOT_MD_USE_MDADM_CONFIG=no
```

Note: The mdadm daemon can send notification events from Software RAID Arrays out via e-mail. The default LifeKeeper installation disables the mdadm. LifeKeeper monitors the array functionality. If mdadm is to be used, it has to be started as a generic application from within the cluster.

An example is given in 7.2.3, “Optional configuration tasks” on page 243

5.4 DB2 Linux, UNIX, and Windows Enterprise Server installation

This section describes the steps necessary to install and configure the DB2 LUW 9.5 Enterprise Server.

We discuss the following topics:

- ▶ Installation of the DB2 software
- ▶ Configuration steps for DB2 on the standby server
- ▶ DB2 instance and database creation with SAPinst
- ▶ Configuring parameters after SAPinst

Information pertaining to prerequisites and certification for DB2 are documented in “DB2 LUW Certification information” on page 95 and “DB2 LUW 9.5 Prerequisites for Linux” on page 95.

5.4.1 Software installation

The DB2 installation media can be acquired from the SAP Marketplace. Place the image into a temporary location on the server and untar the downloaded image to prepare the media.

The DB2 installation binaries can reside on local (internal) or shared (SAN) directories. In the testing for this book, minimal installation was performed using internal drives, and both SAP and DB2 installation resided on SAN volumes.

DB2 V9 offers the advantage of now being able to install multiple versions of DB2 on a single machine. Having multiple copies can make upgrading and patching easier. Older versions of DB2 software required alternate Fix Packs to enable multiple versions within a single server. SAP does not support alternate Fix Packs, so DB2 V9 eases the management of versions.

For the scenario in this book, a shared (SAN) directory rather than a local (internal) directory was used for both the installation binaries and the db2 instance and data. A single installation was performed on the primary cluster server. The db2 user and groups for the instance, the services (/etc/services), and the db2nodes.cfg file were manually adjusted on the failover (secondary) server. Details for the manual steps are outlined subsequently in 5.4.2, “Configuration steps for DB2 on the standby server” on page 134.

For the DB2 client, the installation binaries were also placed on the SAN volume groups. If multiple Application Servers are configured for SAP, care must be taken to ensure the DB2 client is available.

DB2 software on Linux platforms can be installed using either **db2_install**, a response file, or the DB2 setup wizard (db2setup). On the Windows platforms, the command line **db2_install** is not available.

The **db2_install** command line installation installs all components. However, it does not perform user and group creation, instance creation, DAS creation, or configuration.

SAP recommends using the GUI installation process (db2setup) to reduce the number of manual installation steps. The use of the DB2 Setup wizard requires an X Window capable software for rendering the Graphical User Interface.

Installing the software for the DB2 server

To start the DB2 wizard, change to the disk1 subdirectory of the installation image and issue the following command with root authority:

```
./setup
```

It might take a few moments for the launchpad window to appear. Then, for the SAP installation, choose the following options:

- ▶ Select **Install a Product**.
- ▶ Select **DB2 9.5 Enterprise Server Edition** and click **Install New**.
- ▶ Click **Next** and **Accept** the license agreement.
- ▶ Select **Typical installation**.
- ▶ Select the installation directory; `/opt/ibm/db2/V9.5` is the default, as shown in Figure 5-19.

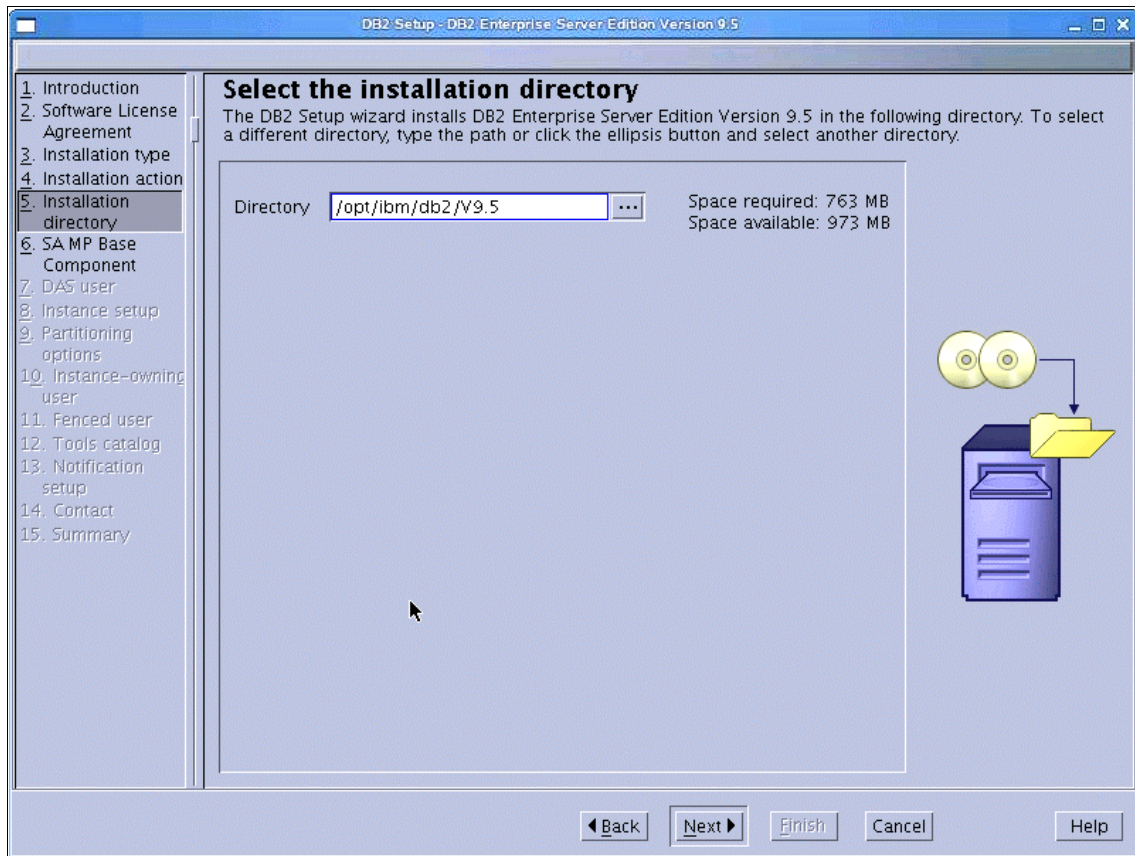


Figure 5-19 Select the installation directory window

- Choose **Do not install SA MP Base Component** as shown in Figure 5-20, because LifeKeeper functions as the software cluster management instead of Tivoli System Automation.

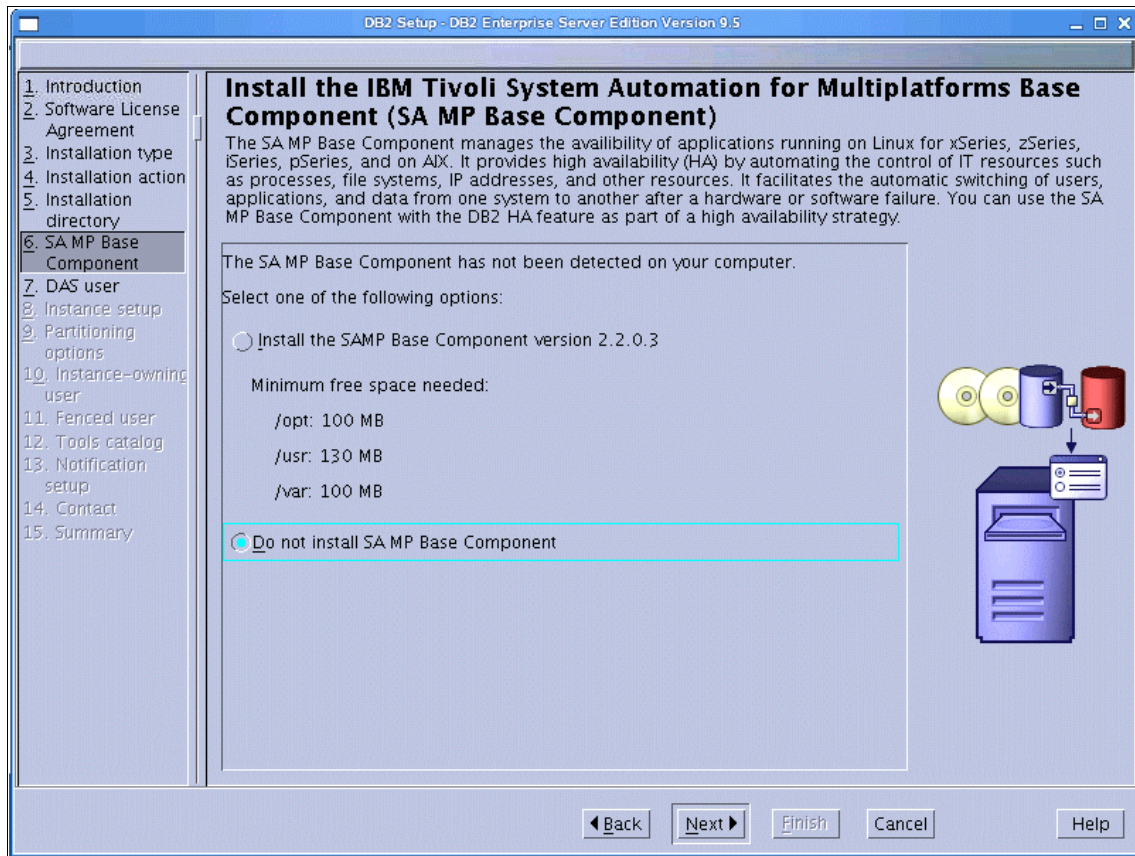


Figure 5-20 Install the IBM Tivoli SA MP component window

- Provide a password and user name for the DB2 Administration Server (DAS).
- On the Instance setup, choose **Do Not create a DB2 instance** as shown in Figure 5-21.

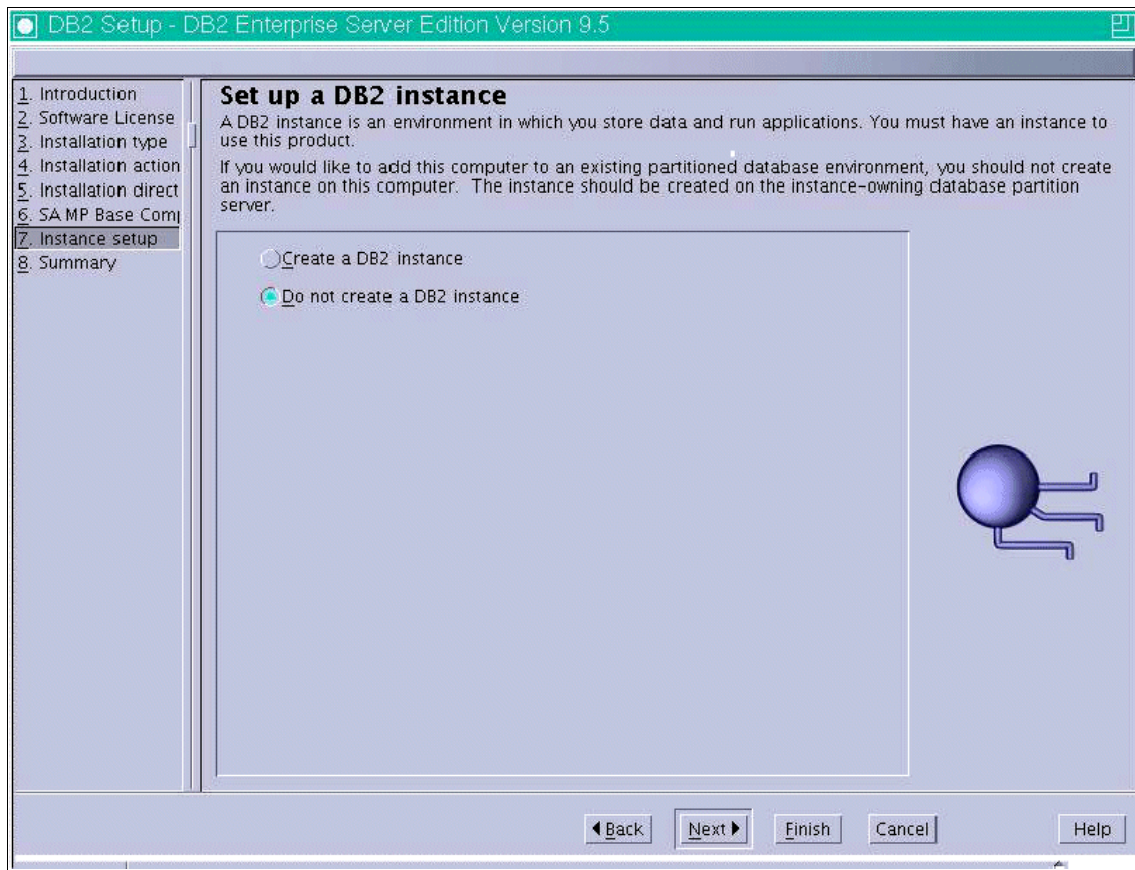


Figure 5-21 Set up a DB2 instance window

- ▶ If an SMTP warning is displayed, ignore the warning.
- ▶ Review the selection and choose **Finish**.

Installing the software for the DB2 client

To install the client software, you follow a process similar to the server install. Change to the directory where the client image is located and issue the following command with root authority:

./setup

It might take a few moments for the launchpad window to appear. Then, for the SAP installation, choose the following options:

- ▶ Select **Install a Product**.
- ▶ Select **Install New** for the DB2 9.5 runtime client.
- ▶ Accept the software license.
- ▶ Select **Typical setup**.
- ▶ Select the location directory for the client installation.
- ▶ Select **Create an instance (client)**.
- ▶ Input the db2 client instance owner and group.
- ▶ Review the selection and choose **Finish**.

After the client software installation completes, a catalog entry for the node and database must be defined. The following example commands are used to define these entries. The host name `sedb` is the virtual name associated with the IP alias for the database services. The service name `sapdb2rdb` is defined in `/etc/services` and also set as the TCP/IP service name in the dbm configuration (`svcname`).

```
db2 catalog tcpip node sedb at sedb server sapdb2rdb
db2 catalog database RDB at node sedb
```

5.4.2 Configuration steps for DB2 on the standby server

Since the installation of the binaries was performed on a shared file system (SAN), it was not necessary to do a second install on the standby server.

However, synchronization of the DB2 user and group must be completed, as well as synchronization of `/etc/services`. Review and modification of `db2nodes.cfg` and ensuring kernel parameters for DB2 IPC resources are the same:

- ▶ Modify `/sqlib/db2nodes.cfg` to reflect the virtual host name as shown:
0 sedb 0
- ▶ Add the DB2 user and group information to the standby server.
- ▶ Add the DB2 ports to `/etc/services` of the standby server.
- ▶ Review and modify the kernel parameters for DB2.
- ▶ Set up user equivalence (`ssh`) or modify `rhosts` if using (`rsh`).

After initial installation, LifeKeeper manages and updates the `db2nodes.cfg` file during both failover and failback scenarios.

5.4.3 DB2 Instance and database creation

The DB2 Instance and database creation is performed with the SAP NetWeaver installation (SAPinst). SAP NetWeaver requires the DB2 binaries to be installed as a prerequisite.

The steps for the SAP NetWeaver installation are given in 5.5.4, “Database installation” on page 142. After the installation is complete, it is necessary to review and modify DB2 database and instance parameters.

During the installation for this book, the SAP NetWeaver installer (SAPinst) encountered an error. SAPinst passed an invalid parameter to the db2icrt command during instance creation. The workaround for this error was to manually create the db2 instance and restart the SAPinst GUI. More detailed information regarding this error is given in Chapter 8, “Troubleshooting” on page 263.

5.4.4 Configuring DB2 settings after SAPinst

After the SAP NetWeaver installation is complete, a review of the DB2 database, database manager (dbm), and registry variables must be performed.

In the implementation for this book, the following tasks were performed:

- ▶ Customize additional database settings (ARCHMETHOD1)
- ▶ Modify the registry to use ssh (default rsh)
- ▶ Execute a full database backup

A production database must be implemented in log archive mode. By default, circular logging is enabled, however for a production database, log archive mode must be enabled.

The SAP installation (SAPinst) modifies the log parameters for active logging. After the installation completes, the archive location must be configured. A backup of the database is required and was performed after the logging parameters were adjusted.

By default, DB2 uses rsh (remote shell) as the communication protocol when starting remote database partitions and remote command used with HA. The registry variable, DB2RSHCMD, can be used to modify the protocol or provide a full path name.

For example, setting this registry variable to the full path name for ssh (secure shell) causes DB2 database products to use ssh as the communication protocol for the requested running of the utilities and commands. It can also be set to the full path name of a script that invokes the remote command program with appropriate default parameters. The instance owner must be able to use the specified remote shell program to log in from each DB2 database node to each other DB2 database node, without being prompted for any additional verification or authentication (that is, passwords or password phrases)

In the installation for this book, the registry variable was modified to use **ssh** instead of the default (**rsh**). Example 5-20 shows the registry variables.

Example 5-20 DB2 registry variables

```
> db2set
DB2_TRUST_MDC_BLOCK_FULL_HINT=YES [DB2_WORKLOAD]
DB2_ATS_ENABLE=YES [DB2_WORKLOAD]
DB2_RESTRICT_DDF=YES [DB2_WORKLOAD]
DB2_SET_MAX_CONTAINER_SIZE=2000000000 [DB2_WORKLOAD]
DB2_OPT_MAX_TEMP_SIZE=10240 [DB2_WORKLOAD]
DB2_WORKLOAD=SAP
DB2_TRUNCATE_REUSESTORAGE=IMPORT [DB2_WORKLOAD]
DB2_MDC_ROLLOUT=DEFER [DB2_WORKLOAD]
DB2RSHCMD=ssh
DB2_SKIPINSERTED=YES [DB2_WORKLOAD]
DB2_VIEW_REOPT_VALUES=YES [DB2_WORKLOAD]
DB2_OBJECT_TABLE_ENTRIES=65532 [DB2_WORKLOAD]
DB2_OPTPROFILE=YES [DB2_WORKLOAD]
DB2_IMPLICIT_UNICODE=YES [DB2_WORKLOAD]
DB2_INLIST_TO_NLJN=YES [DB2_WORKLOAD]
DB2_MINIMIZE_LISTPREFETCH=YES [DB2_WORKLOAD]
DB2_REDUCED_OPTIMIZATION=4,INDEX,JOIN,NO_TQ_FACT,NO_HSJN_BUILD_FACT,STARJN_CARD_SKEW,NO_SORT_MGJOIN,CART OFF,CAP OFF [DB2_WORKLOAD]
DB2NOTIFYVERBOSE=YES [DB2_WORKLOAD]
DB2_INTERESTING_KEYS=YES [DB2_WORKLOAD]
DB2_EVALUNCOMMITTED=YES [DB2_WORKLOAD]
DB2_ANTIJOIN=EXTEND [DB2_WORKLOAD]
DB2ENVLIST=INSTHOME SAPSYSTEMNAME dbs_db6_schema DIR_LIBRARY
LD_LIBRARY_PATH
DB2_RR_TO_RS=YES [DB2_WORKLOAD]
DB2_DROP_NO_WAIT=YES [DB2_WORKLOAD]
DB2COUNTRY=1
DB2COMM=TCPIP [DB2_WORKLOAD]
```

5.5 SAP NetWeaver installation

This section illustrates the installation of the SAP NetWeaver 7.0 system for the test environment used in this book. This is a documented process in SAP manuals, and further information is available at:

<http://service.sap.com/instguidesNW70>

Note: The SAP Marketplace is a secure site and requires having a registered user and password.

5.5.1 Media list

In order to download the correct media kit for your installation is important to check the last version of the media list document, and download the latest available version of the software for your system.

The SAP NetWeaver media used in the test environment for this book are listed in Table 5-6.

Table 5-6 Media list for the test environment

Media Names	Media Numbers
Installation Master NW2004sSR2	51033208_8
DB2UDBOnNW2004sSR2	51033351_2
InstallationExport NW2004sSR2	51032246_1 51032246_2
JavaComponent NW2004sSR2	51032257_1 51032257_2 51032257_3 51032257_4 51032257_5
Kernel NW2004sSR2	51033032_9

At the time of writing this book, there was no DB2 V9.5 available with the NetWeaver 7.0 package. The database software was downloaded according to instructions on SAP Note 101809 available at:

<https://service.sap.com/notes>

Note: The SAP Marketplace is a secure site and requires having a registered user and password.

5.5.2 Steps for installation

The installation of the SAP NetWeaver 7.0 for high availability is achieved in five major steps:

1. Installation of the ABAP and Java Central Services (ASCS and SCS)
2. Installation of the database
3. Installation of the Central Instance
4. Installation of the enqueue replication server
5. Application Server Installation

These steps are shown in Figure 5-22.

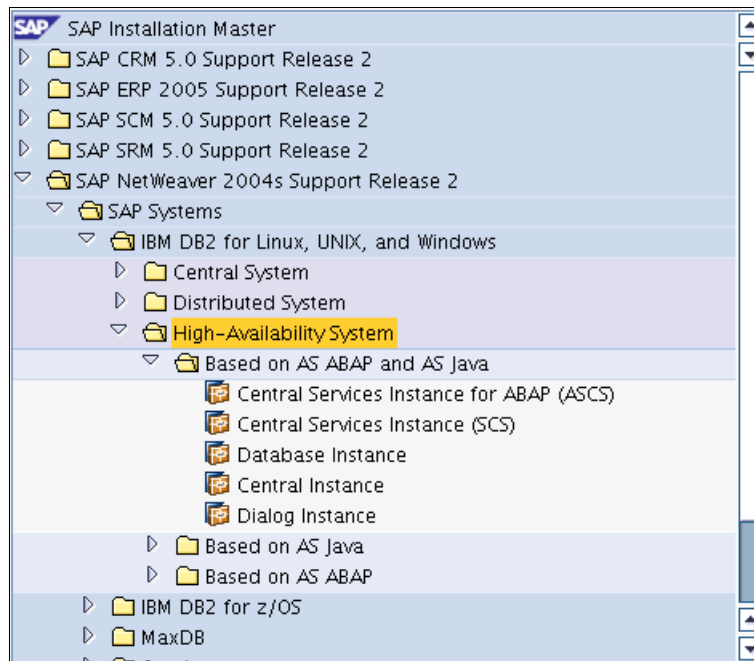


Figure 5-22 High availability installation steps

Before starting SAPinst

SAPinst is the graphical SAP NetWeaver 7 installation tool. Detailed information about how to use it can be obtained in the SAP Installation manuals.

Before starting SAPinst, make sure that the requirements described in 5.1, “Prerequisites” on page 90 are met.

The operating system users for the database and SAP systems can be created before starting the installation, but SAPinst creates them if not yet present in the operating system. Unless there are special requirements, it is less error prone to let SAPinst deal with them.

Set the following environment variables in the installation session as listed in Table 5-7.

Table 5-7 Variables used in the book test environment installation

Name	Meaning
<code>SAPINST_JRE_HOME</code>	This is the location of Java binaries for SAPinst.
<code>TEMP</code>	SAPinst looks for the variables <code>TEMP</code> , <code>TMP</code> , and <code>TMPDIR</code> . If no values are set for these variables, sapinst uses <code>/tmp</code> as its temporary directory. Ensure that at least 200 MB is available for SAPinst in the temporary directory.
<code>SAPINST_USE_HOSTNAME</code>	For a high availability installation, the virtual host name of your cluster must be set in this variable. Otherwise, use the appropriate option of SAPinst command.

Table 5-8 describes the three options for displaying the SAPinst window.

Table 5-8 Installation display options

Option	How it works
Local	SAPinst starts the GUI in the same server where the installation is running.
Remote	SAPinst starts the GUI in the remote server specified in the variable <code>DISPLAY</code> .
Remote SAPinst	SAPinst is started only as a server, and an SAPinst client connects to this server.

For further details regarding the installation options, check the SAP installation guide for your operating system and database combination.

For the test environment for this book, the following options were set, as shown in Example 5-21.

Example 5-21 Setting variable for installation

```
se01:~ # export SAPINST_JRE_HOME=/opt/IBMJava2-amd64-142/  
se01:~ # export SAPINST_USE_HOSTNAME=sesap  
se01:~ # export TEMP=/install/sapinst
```

To start SAPinst, execute the **sapinst** program from the Master DVD (Installation Master NW2004sSR2) as in Example 5-22 used in the book test environment.

Example 5-22 Starting sapinst interface

```
se01:/ # /install/BS_2005_SR2_SAP_Installation_Master/IM_LINUX_X86_64/sapinst
```

5.5.3 SAP Central Services installation

In a switch-over installation, along with the installation of the central services for the Java Application Server, there is the installation of the ABAP System Central Services (ASCS). This is necessary because these services are single points of failure (SPOFs) and therefore have to be protected by the cluster software.

The ABAP Central Services is the first SAP software to be installed in a cluster solution with ABAP and Java usage types. To learn more about usage types, consult the *SAP NetWeaver 7.0 Master Guide*, Material number: 50076017 available at:

<https://service.sap.com/instguides>

The SAP Central services for the Java Web Application Server is installed only if the usage type of your SAP system requires it. For the test scenario of this book, we selected a usage type (SAP PI) that requires both ABAP and Java Application Servers.

Installing the SAP software

Select the appropriate installation case under the **High-Availability** option. For the example used in this book, we used **Based on AS ABAP and AS Java → Central Services Instance for ABAP (ASCS)** as shown in Figure 5-23.

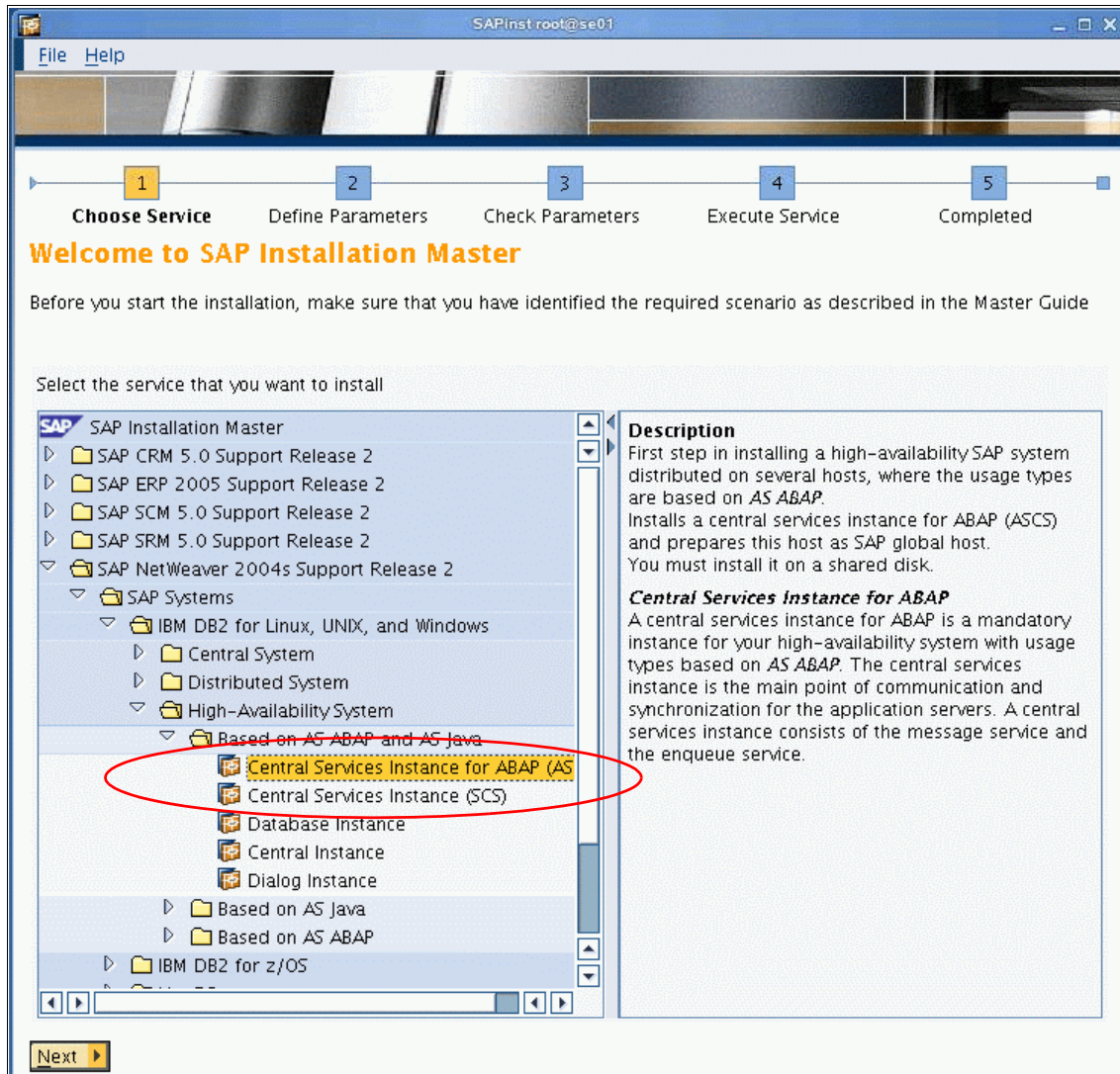


Figure 5-23 Installing central services for ABAP and Java

Click the **Next** button to proceed and enter the required input parameters. To find more information on each parameter during the input phase of the installation, position the cursor on the field of the respective parameter and press F1. After you have entered all requested input parameters, SAPInst displays the Parameter Summary window. At this point you can still change any allowed parameter value by clicking the **Revise** button.

After installing the Central Services Instance for ABAP, execute **sapinst** again and install the Central Services Instance component.

5.5.4 Database installation

The database used in this example installation was the DB6, which is the SAP name of DB2 for Linux, UNIX, and Windows.

According to the SAP installation guide, *SAP NetWeaver 7.0 SR2 ABAP+Java on Linux: IBM DB2 Universal Database for UNIX and Windows*, it is recommended to use the latest version of the database available.

The installation guides are available at:

<http://service.sap.com/instguides>

Note: The SAP Marketplace is a secure site and requires having a registered user and password.

Although each database has its particular installation procedures, the differences from a SAP NetWeaver point of view are minimal. Check the installation guide for your database for further information.

The database installation option in the SAPinst window assumes that the database software is already installed, except for Oracle. For Oracle databases, SAPinst stops the installation and inform you to install the database software.

During this phase, SAPinst creates all the logical structure of the database:

- ▶ Database instance
- ▶ Database
- ▶ ABAP and Java schemes

After the logical structure of the database is created, SAPinst imports ABAP and Java data from InstallationExport NW2004sSR2 and JavaComponent NW2004sSR2.

The import process can be followed in the SAPinst window and also in the **ImportMonitor.console.log** (for ABAP) and **import_monitor.java.log** (for Java) under SAPinst installation directory. Typical output from these files is shown in Example 5-23.

Example 5-23 ImportMonitor.console.log file output

```
Import Monitor jobs: running 1, waiting 17, completed 0, failed 0, total 18.
Loading of 'SAPSDIC' import package: OK
Import Monitor jobs: running 0, waiting 17, completed 1, failed 0, total 18.
Import Monitor jobs: running 1, waiting 16, completed 1, failed 0, total 18.
Import Monitor jobs: running 2, waiting 15, completed 1, failed 0, total 18.
Import Monitor jobs: running 3, waiting 14, completed 1, failed 0, total 18.
Import Monitor jobs: running 4, waiting 13, completed 1, failed 0, total 18.
Loading of 'SAPAPPL1' import package: OK
Import Monitor jobs: running 3, waiting 13, completed 2, failed 0, total 18.
Import Monitor jobs: running 4, waiting 12, completed 2, failed 0, total 18.
```

As a rule of thumb, define 2 parallel process per processor core. The definition of this value depends on the available hardware capacity.

5.5.5 Central Instance installation

Central instance installation is a required step both for Java and ABAP stacks. Since it is no longer a single point of failure, it does not have to be protected by the cluster software.

Although the name of the component is still Central Instance, it is no different than any other SAP Application Server, which means that this component can be replicated to achieve high availability. By default, the SDM is installed with the Central Instance, but since it is used for software deployment, this might not be an issue in production systems and was not commented upon in this book.

The installation of this component group follows the same pattern of central services. Select the option as shown in Figure 5-24. Select **Next** and enter the input parameter requested.

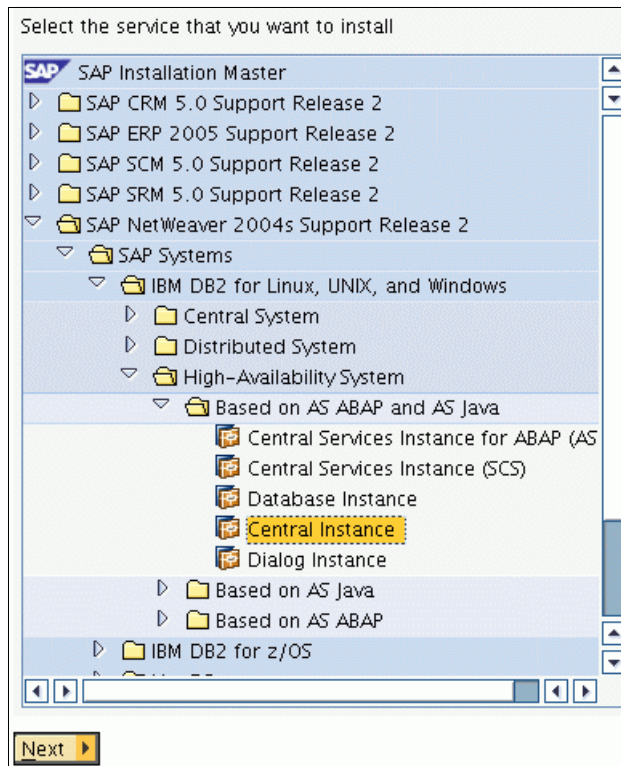


Figure 5-24 Central Instance installation

5.5.6 Installation of the enqueue replication server

The installation of the enqueue replication server is a necessary step to ensure that the enqueue server is protected against failures. The enqueue replication server keeps a copy of the original enqueue table in its own shared memory. When an enqueue server fails, a new one can be started with this replicated copy. The SAP transaction no longer needs to be reset in an event of enqueue server failure.

At the time of writing this book, it was not possible to install an enqueue replication server using SAPinst for UNIX and Linux platforms. However, in SAP NetWeaver 7.0, the installation of the SAP central services already installs a standalone enqueue server, which is a necessary step to setting up the enqueue replication server.

To install the replication for two servers for the environment in this book, as shown in Figure 5-25, complete the following steps:

1. Create the directory structure.
2. Copy files between executable directories.
3. Create an SAP copy files list.
4. Create start and instance profiles for the enqueue replication server.
5. Configure the control mechanism.

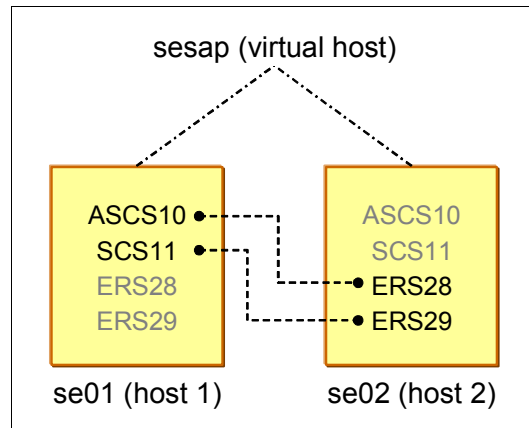


Figure 5-25 Enqueue Replication Server topology

Create the directory structure

A directory structure similar to the one shown in Figure 5-26 must be created in both servers using the SAP administration user (<sid>adm).

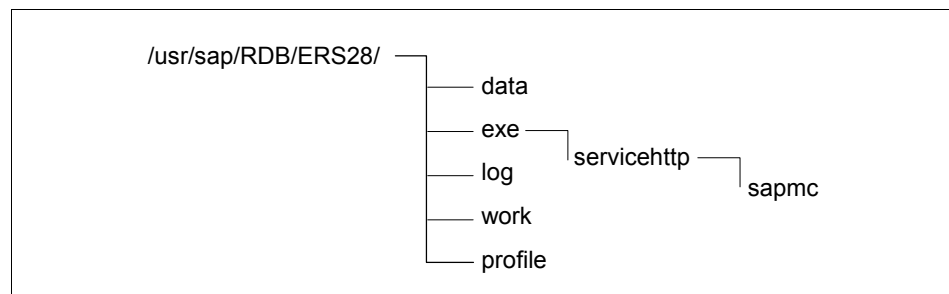


Figure 5-26 Directory structure for an enqueue replication instance

The binaries directory of the enqueue replication instance should be the same one as the enqueue server. This directory is defined by the DIR_CT_RUN variable in the instance profile.

With the cluster solution for the test environment, the `/sapmnt/RDB/profile` was in the shared drive. In this case it was necessary to create the directory `/usr/sap/RDB/ERS28/profile` on both servers.

Copy files between executable directories

Copy the following files from the instance executable directories to the enqueue replication directory as listed in Table 5-9.

Table 5-9 File copies to ERS28 directory structure

From: /sapmnt/RDB/exe	To: /usr/sap/RDB/ERS28/exe
enqt enrepserver ensmon libicudata.so.30 libicui18n.so.30 libicuuc.so.30 libsapu16_mt.so librfcum.so sapcpe sapstart sapstartsrv sapcontrol	
From: /sapmnt/RDB/exe/servicehttp/sapmc	To: /usr/sap/RDB/ERS28/exe/servicehttp/ sapmc
sapmc.jar sapmc.html frog.jar soapclient.jar	

Create a list file

Create an SAP copy list file under the enqueue replication server executable directory with the file names shown in Example 5-24.

Example 5-24 SAP copy list file

```
se01:/usr/sap/RDB/ERS28/exe # cat er1.lst
enrepserver
ensmon
enqt
libsapu16_mt.so
libsapu16.so
libicuuc.so.30
```

```
libicudata.so.30
libicu18n.so.30
librfcum.so
sapcpe
sapstartsrv
sapstart
sapcontrol
servicehttp
ers.lst
se01:/usr/sap/RDB/ERS28/exe #
```

Create start profiles and instance profiles

For every enqueue replication server created, it is necessary to create a start profile and an instance profile per host.

The start profile for the test environment of this book is shown in Example 5-25.

Example 5-25 Enqueue server start profile

```
SAPSYSTEMNAME = RDB
SAPSYSTEM = 28
INSTANCE_NAME = ERS28
DIR_CT_RUN = $(DIR_EXE_ROOT)/run
DIR_EXECUTABLE = $(DIR_INSTANCE)/exe
SAPLOCALHOST = se01
DIR_PROFILE = $(DIR_INSTANCE)/profile
_PF = $(DIR_PROFILE)/RDB_ERS28_se01
SETENV_00 = LD_LIBRARY_PATH=$(DIR_LIBRARY):%(LD_LIBRARY_PATH)
SETENV_01 = SHLIB_PATH=$(DIR_LIBRARY):%(SHLIB_PATH)
SETENV_02 = LIBPATH=$(DIR_LIBRARY):%(LIBPATH)
#-----
# Copy SAP Executables
#-----
_CPARGO = list:$(DIR_CT_RUN)/ers.lst
Execute_00 = immediate $(DIR_CT_RUN)/sapcpe$(FT_EXE) pf=$(_PF)
$(_CPARGO)

#-----
# start enqueue replication server
#-----
_ER = er.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_01 = immediate rm -f $(_ER)
Execute_02 = local ln -s -f $(DIR_EXECUTABLE)/enrepserver $(_ER)
Restart_Program_00 = local $(_ER) pf=$(_PF) NR=$(SCSID)
```

The key parameters of this file were highlighted. Observe where the SAP copy list file is being used.

The instance profile is shown in Example 5-26.

Example 5-26 Replication enqueue instance profile

```
SAPSYSTEMNAME = RDB
SAPSYSTEM = 28
INSTANCE_NAME = ERS28
DIR_CT_RUN = $(DIR_EXE_ROOT)/run
DIR_EXECUTABLE = $(DIR_INSTANCE)/exe
DIR_PROFILE = $(DIR_INSTANCE)/profile
SAPLOCALHOST = se01

#-----
# Settings for enqueue monitoring tools (enqt, ensmon)
#-----
enqueue/process_location = REMOTESA
rdisp/enqname = $(rdisp/myname)

#-----
# standalone enqueue details from (A)SCS instance
#-----

SCSID = 10
SCSHOST = sesap

enqueue/serverinst = $(SCSID)
enqueue/serverhost = $(SCSHOST)

#-----
# replica table file persistency
#-----
```

Changes in DEFAULT.PFL and enqueue server instance profile

Additionally, it is necessary to insert the following parameters into the enqueue central services being replicated, as shown in Example 5-27.

Example 5-27 Parameter for SAP central services instance profiles

```
enqueue/server/replication = true
```

Do the same for *DEFAULT.PFL* profile as shown in Example 5-28.

Example 5-28 Change on DEFAULT.PFL profile file

```
enqueue/dequeue_wait_answer = TRUE
```

Configure the control mechanism

After installing the enqueue replication server, it is necessary to configure a control mechanism to provide the automatic start and stop when necessary.

LifeKeeper for Linux provides mechanisms to check if the enqueue server is running or not, so it activates or deactivates the enqueue replication server. The enqueue replication server protection is not an extra resource in the LifeKeeper hierarchy. The protection is included in the LifeKeeper resource for the SAP Central Services. The protection is enabled by a control file in the LifeKeeper directory `/opt/LifeKeeper/subsys/appsuite/resources/sap` described in 5.8.5, “Creating SAP resources” on page 182. The LifeKeeper SAP hierarchy is shown in Figure 5-27.

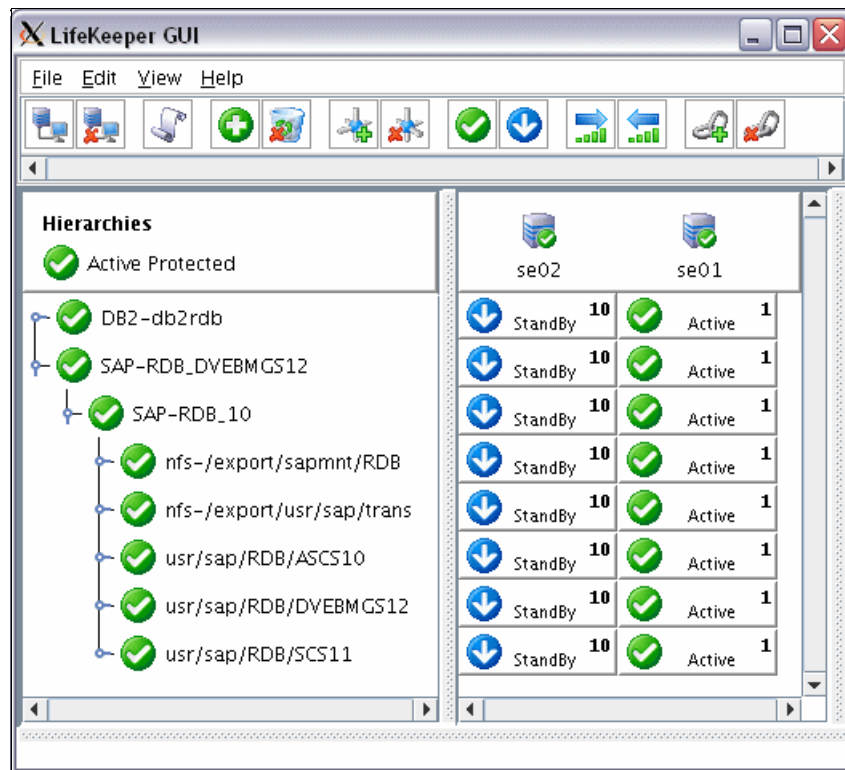


Figure 5-27 SAP hierarchy

This control can rather be performed by a polling script together with some parameters in the instance profile. For further information regarding the setting up of the enqueue replication server, check the following link:

http://help.sap.com/saphelp_nw2004s/helpdata/en/de/cf853f11ed0617e1000000a114084/content.htm

5.5.7 Application Server installation

Although the installation of the SAP Application Server (previously called a Dialog Instance) is an optional step, for a high available system it is mandatory. Having additional Application Servers on a high availability system ensures that the ABAP and Java are replicated.

The Application Server is still referred as a Dialog Instance in the installation guide and SAPinst tool, however according to the *Technical Infrastructure Guide - SAP NetWeaver 2004s*, it is now called an Application Server. The guide can be downloaded from the SAP Service Marketplace at:

<https://service.sap.com/installnw70>

An important requirement, in order to install the Application Server, is the installation of the database client before starting SAPinst. Consult the installation guide for your database for further details.

In the test scenario for this book, an Application Server was installed in each host so that, while the central instance was running in one host, the application server is running in another, providing the same services.

Invoke the SAPinst as explained in 5.5.2, “Steps for installation” on page 138 and mainly in 5.5.3, “SAP Central Services installation” on page 140 by executing the **sapinst** program from the Master DVD as in Example 5-22 on page 140.

Select the Dialog Instance service located under the High Availability option from the installation master panel and click **Next** to proceed and enter the required input parameters. Figure 5-28 shows the Application Server (Dialog Instance) installation completion window.

Figure 5-28 shows the Application Server installation completion window.

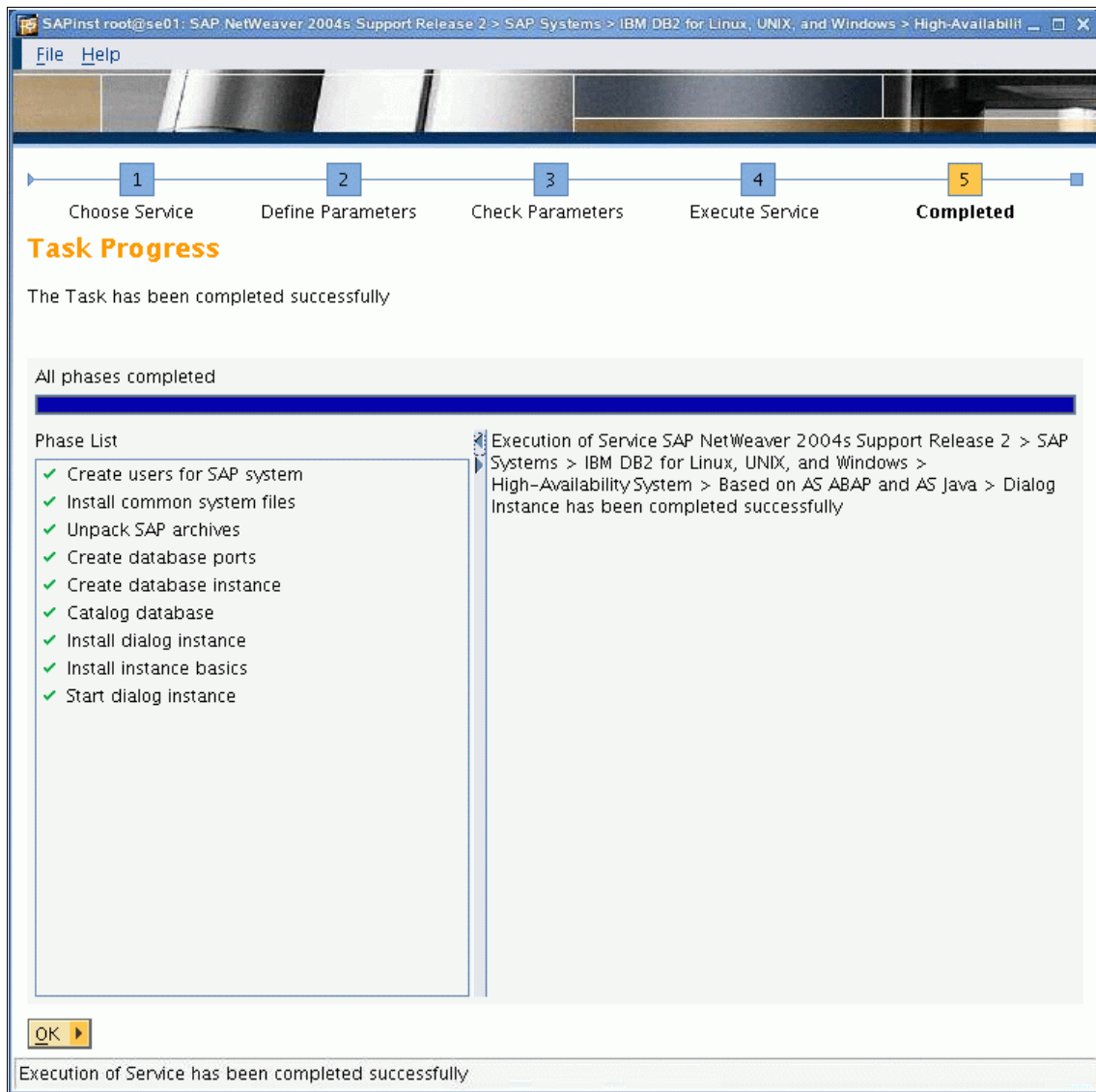


Figure 5-28 SAP Application Server installation

5.6 LifeKeeper cluster software installation

This section describes the installation of the LifeKeeper software.

LifeKeeper software is divided into three parts:

- ▶ LifeKeeper Installation Support CD
- ▶ LifeKeeper Core package
- ▶ Necessary Application Recovery Kits

5.6.1 LifeKeeper Support CD installation

The LifeKeeper for Linux Installation Support CD provides a set of installation scripts designed to perform user-interactive system setup tasks that are necessary before LifeKeeper can be installed on your system. The Installation Support CD identifies what Linux distribution is running and, through a series of answers provided, installs various packages required to ensure a successful LifeKeeper installation. It also installs a licensing utilities package that provides utilities for obtaining and displaying the host ID of your server. Host IDs are used to obtain a valid license key for running LifeKeeper.

The LifeKeeper Installation Support CD is distributed as an iso image. To use the iso image, mount the CD as root user with the following command:

```
mount -o loop,ro de.img /media/cdrom
```

Start the installation by following these steps:

1. Change to the CDROM directory and run the following command:

```
sh ./setup
```

2. Text is displayed, explaining what is going to occur during the installation procedure. There are a series of prompts that require an answer, **y** for Yes or **n** for No. The type and sequence of the questions are dependent on the Linux distribution.

Read each question carefully to ensure a proper response. It is recommended that the answer is Yes to each question in order to complete all the steps required for a successful LifeKeeper Installation.

3. The last item in the setup script is the installation of the LifeKeeper licensing utilities. See the section “Obtaining and Installing the license key” on page 153 for details.
4. After all the questions posed by the setup script have been answered, it informs you that the installation was successful.
5. Finally, reboot the system in order to incorporate the kernel with new modules that were just installed. Then continue to install the LifeKeeper RPM packages from the LifeKeeper Core.

Note: LifeKeeper core RPMs are located on the Core CD image.

Obtaining and Installing the license key

LifeKeeper requires a unique license key for each server for the Core and each optional recovery kit. The license must be installed before LifeKeeper can be started and run successfully.

The Installation Support script installs the Licensing Utilities package, which obtains and displays the host ID of your server. The host ID, along with the authorization code that was provided with your LifeKeeper software, is used to obtain permanent license keys required to run LifeKeeper.

Perform the following steps to obtain and install the license key for each server in the LifeKeeper cluster:

1. Make note of the host ID displayed by the licensing utility in the Installation Support setup script. If you have to obtain your host ID again at a later time, use the command `/opt/LifeKeeper/bin/lmhostid`.
2. The setup script from LifeKeeper Support CD asks if the license keys have to be installed.
3. Obtain license keys at:

<http://www.steeleye.com/support>

Click the **License Key** link. After logging in when prompted, enter your host ID and authorization code for each software license needed. Immediately after providing the necessary information, each license key is sent in a separate e-mail.

Note: The license key can be a long string. Therefore, in the mail received, Edit-Copy each license key and then paste it when prompted by the license key utility (see the steps below). Otherwise, we recommend that you save the license keys in a text file (each key on a separate line) and copied to the LifeKeeper server. The license keys can be read directly from the file by the `lkkeyins` utility.

4. Answering **Yes** in step 2 starts the `lkkeyins` utility automatically.
5. Answering **No** ends the setup script, and the license keys can be installed later by running the command `/opt/LifeKeeper/bin/lkkeyins`. Enter the license key using one of the following methods:
 - Cut and paste the string from the mail file.
 - Enter a filename containing the license key obtained by mail.
 - Type the string manually.

6. To verify the installed license keys, use the command
/opt/LifeKeeper/bin/typ_list -l -f:
There must be no lines with “NEEDED” listed in the output. All lines must either be “UNLICENSED” or “PERMANENT.”
7. Repeat these steps for all servers in the cluster.

5.6.2 LifeKeeper software installation

Install the LifeKeeper software on each server in the LifeKeeper configuration. The LifeKeeper Core is installed first, followed by the optional recovery kit software.

To install the LifeKeeper core software, use the following command:

```
rpm -ihv <package name>.<architecture>.rpm
```

The following core packages are installed:

- ▶ steeleye-lk-<LifeKeeper Version>
- ▶ steeleye-lkGUI-<LifeKeeper Version>
- ▶ steeleye-lkHLP-<LifeKeeper Version>
- ▶ steeleye-lkIP-<LifeKeeper Version>
- ▶ steeleye-lkMAN-<LifeKeeper Version>
- ▶ steeleye-lkRAW-<LifeKeeper Version>

Next, install the additional recovery kit packages:

- ▶ steeleye-lkLVM-<Version>.noarch.rpm
- ▶ steeleye-lkNAS-<Version>.noarch.rpm
- ▶ steeleye-lkNFS-<Version>.noarch.rpm
- ▶ steeleye-lkSAP-<Version>.noarch.rpm
- ▶ Recovery Kit for the used Database:
 - steeleye-lkDB2-<Version>.noarch.rpm
 - steeleye-lkORA-<Version>.noarch.rpm
 - steeleye-lkSAPDB-<Version>.noarch.rpm
- ▶ Additional Recovery Kits example for Multipath environment and/or Software RAID (md):
 - steeleye-lkDMMP-<Version>.noarch.rpm
 - steeleye-lkMD-<Version>.noarch.rpm

Attention: Installing LifeKeeper on your shared storage is not supported. Each server must have its own copy installed on its local disk.

By default, LifeKeeper packages are installed in the directory `/opt/LifeKeeper`. Using this default directory is recommended.

For details about relocating the LifeKeeper package, refer the *LifeKeeper for Linux v6 Planning and Installation Guide*, available at:

<http://www.steeleye.com/support>

5.7 Cluster configuration

This section provides information for starting the LifeKeeper server daemon processes and setting up the cluster.

After all installations tasks completed, the LifeKeeper has to start on both servers.

Attention: The network configuration of all cluster nodes must be verified carefully prior to starting the resources. Ensure that all nodes resolve to the same IP addresses and back; all nodes are able to connect to each other. Ensure that the IP addresses are assigned statically and host names are put into `/etc/hosts`, independent of any DNS. This prevents possible errors.

5.7.1 Starting LifeKeeper

LifeKeeper provides a command line interface that starts and stops the LifeKeeper daemon processes. These daemon processes must be running before the LifeKeeper Graphical User Interface (GUI) can be started.

If LifeKeeper is not currently running on the system, type the following command as the user root on all servers:

`/opt/LifeKeeper/bin/lkstart`

Following a delay of a few seconds, a message similar to the one shown in Example 5-29 is displayed.

Example 5-29 Start message of LifeKeeper

```
se01:/opt/LifeKeeper/config # lkstart
```

```
steel-eye-lk
```

```
6.2.1
```

```
Copyright (C) 2000-2008 SteelEye Technology Inc.
```

```
(suse 10)
```

```
LIFEKEEPER STARTING TO INITIALIZE AT: Thu Feb 28 11:16:48 CET 2008
```

```
LifeKeeper is starting to initialize at Thu Feb 28 11:16:48 CET 2008
```

```
LIFEKEEPER NOW RUNNING AT: Thu Feb 28 11:17:11 CET 2008
```

This command modifies */etc/inittab* to include all LifeKeeper daemon processes.

Starting the LifeKeeper GUI server

The LifeKeeper GUI uses Java technology to provide a Graphical User Interface to LifeKeeper and its configuration data. Because the LifeKeeper GUI is a client/server application, a user runs the Graphical User Interface on a client system in order to monitor or administer a server system where LifeKeeper is running.

The LifeKeeper GUI server is initialized on each server in a LifeKeeper cluster at system startup. It communicates with the LifeKeeper core software via the Java Native Interface and with the LifeKeeper GUI client using Hypertext Transfer Protocol and Remote Method Invocation.

First, the LifeKeeper GUI server must be started by typing the following command as the user root on all servers:

```
/opt/LifeKeeper/bin/lkGUIserver start
```

A message as shown in Example 5-30 is displayed.

Example 5-30 Start message of GUI server

```
# Installing GUI log
```

```
# LifeKeeper GUI Server Startup at:
```

```
#      Mon Feb 25 17:04:53 CET 2008
```

```
# Setting up inittab entries
```

```
# LifeKeeper GUI Server Startup Completed at:
```

```
#      Mon Feb 25 17:04:53 CET 2008
```

This command modifies */etc/inittab* to include the GUI server.

Note: After the GUI Server has been started following an initial installation, starting and stopping LifeKeeper also starts and stops all LifeKeeper daemon processes, including the GUI server.

LifeKeeper GUI server configuration

The LifeKeeper GUI server includes three classes of GUI users, with different permissions for each:

- ▶ Users with Administrator permission throughout a cluster can perform all possible actions through the GUI.
- ▶ Users with Operator permission on a server can view LifeKeeper configuration and status information, and can bring resources into service and take them out of service on that server.
- ▶ Users with Guest permission on a server can view LifeKeeper configuration and status information on that server.

During installation of the GUI package, an entry for the root login and password is automatically configured in the GUI password file with Administrator permission, allowing root to perform all LifeKeeper tasks on that server via the GUI application or Web client. If the plan is to allow users other than root to use LifeKeeper GUI clients, then LifeKeeper has to be configured for GUI users.

User administration is performed through the command line interface, using **lkpasswd**:

- ▶ To grant a user Administrator permission for the LifeKeeper GUI, type the following command:
/opt/LifeKeeper/bin/lkpasswd -administrator <user>
- ▶ To grant a user Operator permission for the LifeKeeper GUI, type the following command:
/opt/LifeKeeper/bin/lkpasswd -operator <user>
- ▶ To grant a user Guest permission for the LifeKeeper GUI, type the following command:
/opt/LifeKeeper/bin/lkpasswd -guest <user>
- ▶ To change the password for an existing user without changing their permission level, type the following command:
/opt/LifeKeeper/bin/lkpasswd <user>
- ▶ To prevent an existing user from using the LifeKeeper GUI, type the following command:
/opt/LifeKeeper/bin/lkpasswd -delete <user>

Note: These commands update the GUI password file only on the server being administered. Repeat the command on all servers in the LifeKeeper cluster.

After completing all of these tasks, the LifeKeeper GUI can be started.

LifeKeeper GUI client

LifeKeeper ships with two clients that can connect to the GUI server:

- ▶ An application client, which is designed to run on Linux systems
- ▶ A Web client, which can be run from any system that can connect to ports 81 and 82 of all servers in the cluster.

Both LifeKeeper clients include the same graphical components.

Running the GUI on a LifeKeeper server

The simplest way to run the LifeKeeper GUI is as an application on a LifeKeeper server. By doing so, the GUI client and server are running on the same system. To do so, as user root, type the following command:

```
/opt/LifeKeeper/bin/lkGUIapp
```

The lkGUIapp script sets the appropriate environment variables and starts the application. As the application is loading, an application identity dialog or “splash” window for LifeKeeper is presented.

After the application is loaded, the LifeKeeper GUI is presented and the Cluster Connect dialog is automatically displayed. Enter the Server Name you want to connect to, followed by the login and password.

When a connection to the cluster is established, the GUI window displays a visual representation and status of the resources protected by the connected servers.

Running the GUI on a remote system

Another possibility is to run the LifeKeeper GUI as a Java applet in a browser. To do so, start the browser on the remote machine and type the following address in the address field, where <server name> is the name of the LifeKeeper server:

```
http://<server name>:81
```


When the LifeKeeper GUI Web page is opened, the following actions take place:

1. The splash window is displayed.
2. The applet is loaded.
3. The Java Virtual Machine is started.
4. Some server files are downloaded.
5. The applet is initialized.

If everything loads properly, a **Start** button is now shown in the applet area. When prompted, click **Start**. The LifeKeeper GUI is presented, and the Cluster Connect dialog is automatically displayed. Enter the Server Name you want to connect to, followed by the login and password.

When a connection to the cluster is established, the GUI window displays a visual representation and status of the resources protected by the connected servers.

Note: The client must resolve the server name in the short form returned by `/opt/LifeKeeper/bin/sys_list`. This can involve adding domain names to `/etc/resolv.conf` on UNIX or the domain search list on Linux computers.

If the splash window does not display a Start button or if there are other problems during the start of the LifeKeeper GUI, refer to the Applet Troubleshooting section, or see the GUI Network-Related Troubleshooting sections in the *LifeKeeper for Linux v6 Planning and Installation Guide* available at:

<http://www.steeleye.com/support>

5.7.2 Creating a cluster configuration

Before LifeKeeper protection can be activated, the communications path (heartbeat) definitions must be created within LifeKeeper. It is essential that the GUI is connected to both (or all) of the servers that you want to add to your LifeKeeper cluster before trying to create a communications path.

To create a communications path between a pair of servers, define the path individually on both servers. LifeKeeper allows you to create both TCP (TCP/IP) and TTY communications paths between a pair of servers. However, you can create multiple TCP communications paths between a pair of servers by specifying the local and remote addresses that are to be the end-points of the path. A priority value is used to tell LifeKeeper the order in which TCP paths to a given remote server should be used. Only one TTY path can be created between a given pair.

Note: A TTY comm path is cluster communication occurring through RS-232 style serial ports and a null modem cable. This feature is not getting used a lot in LifeKeeper today, because many new servers do not have serial ports any longer. This applies to Blade servers.

Using a single communication path can potentially compromise the ability of servers in a cluster to communicate with one another. If a single communication path is used and the communication path fails, LifeKeeper hierarchies can come in-service on multiple servers simultaneously. This is known as *false failover*. Additionally, heavy network traffic on a TCP communication path can result in unexpected behavior, including false failovers and LifeKeeper initialization problems.

To create the communication paths, login to the LifeKeeper GUI as an Administrator user as shown in Figure 5-29.

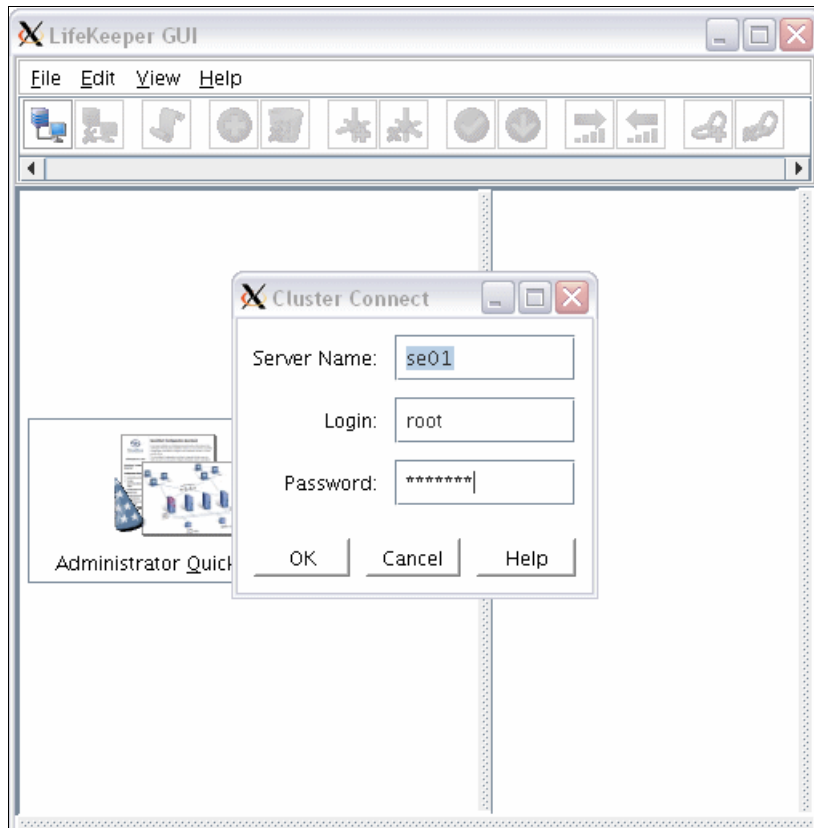


Figure 5-29 Cluster Connect

1. There are four ways to start creating communication paths:
 - Right-click on a server icon, then click **Create Comm Path** when the server context menu is presented, as shown in Figure 5-30.

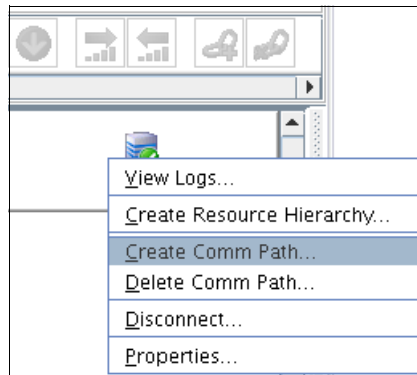


Figure 5-30 Server context menu

- On the global toolbar, click the **Create Comm Path** button as shown in Figure 5-31.

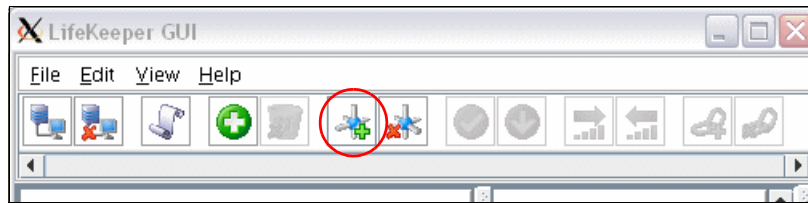


Figure 5-31 Toolbar

- On the **Edit** menu, select **Server**, then **Create Comm Path** as shown in Figure 5-32.

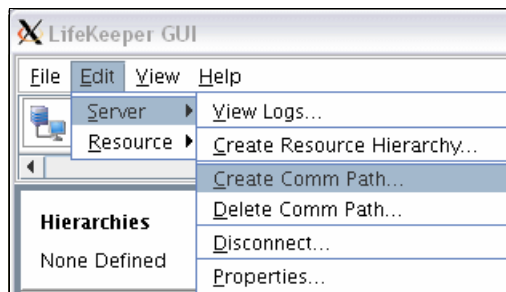


Figure 5-32 Edit menu

- On the server context toolbar, if displayed, click the **Create Comm Path** button as shown in Figure 5-33.

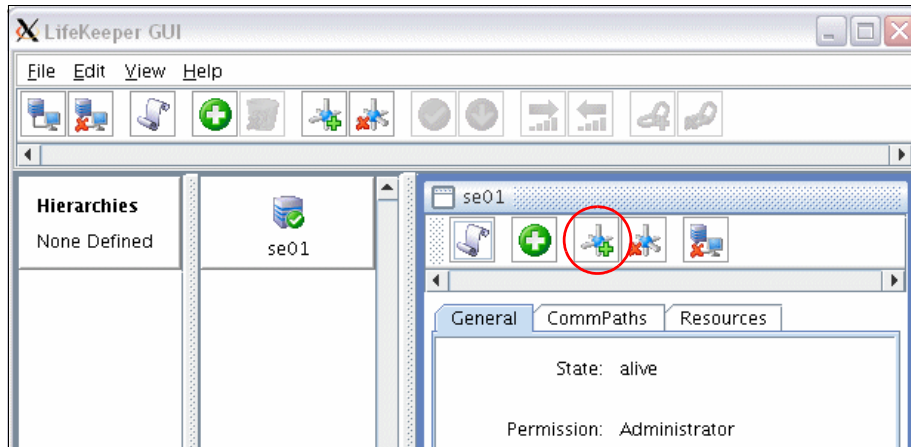


Figure 5-33 Server context toolbar

A wizard is started and directs you through all steps as required.

2. If it is necessary, select the local server from the list box.
3. Select the remote server in the list box as shown in Figure 5-34. If a remote server is not listed in the list box (this means that the server is not yet connected to the cluster), the server name can be entered using **Add**. Ensure that the network addresses for both the local and remote servers are resolvable.

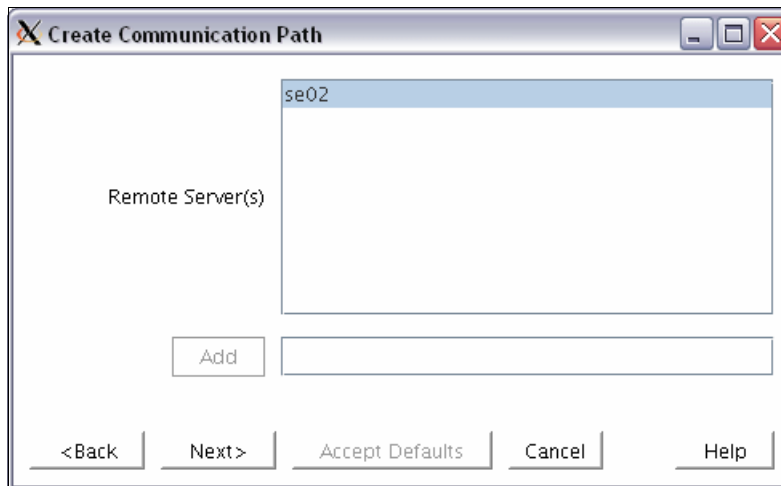


Figure 5-34 List box Remote Server

4. Select either **TCP** or **TTY** for device type from the list box, as shown in Figure 5-35.

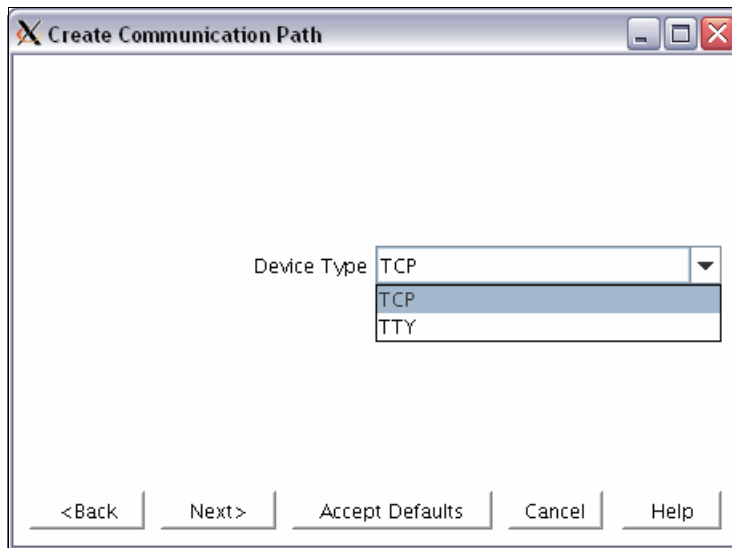


Figure 5-35 Select box Device Type

5. Select one or more local IP addresses if the device type was set for TCP, as shown in Figure 5-36. Select the local TTY device if the device type was set to TTY.

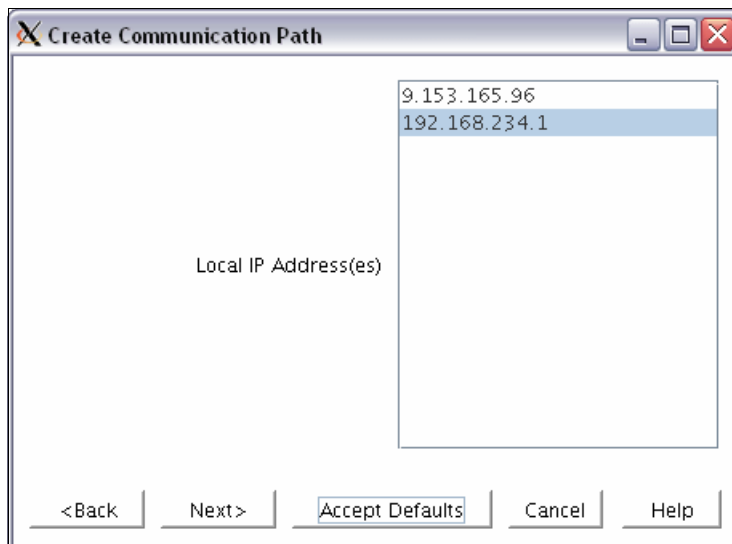


Figure 5-36 List box of local IP addresses

6. Select the remote IP address if the device type was set for TCP. Select the remote TTY device if the device type was set for TTY.
7. Enter the priority for this communication path if the device type was set to TCP. Select the baud rate for this communication path if the device type was set for TTY.
8. Create the prepared communication path. A message should be displayed indicating the network connection is successfully created as shown in Figure 5-37. Click **Next**.

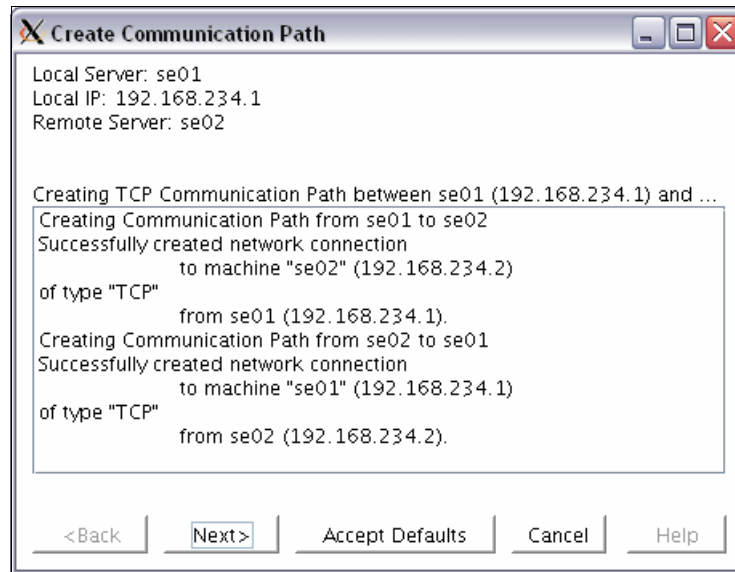


Figure 5-37 Message box

9. If multiple local IP addresses were selected in step 5, then all steps beginning from step 6 must be repeated with the next communication path.
10. If this is the first comm path that has been created, the server icon shows a yellow heartbeat, indicating that one communication path is ALIVE, but there is no redundant communication path, as shown in Figure 5-38.

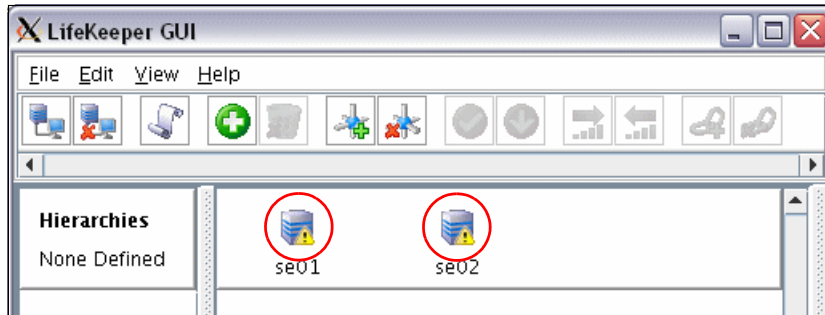


Figure 5-38 Server icons indicating non redundant communication path

The server icon displays a green heartbeat when there are at least two comm paths ALIVE as shown in Figure 5-39.

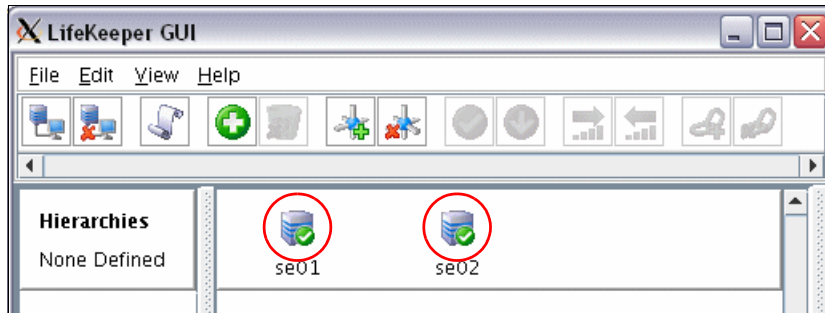


Figure 5-39 Server icons indicating redundant communication paths

In these steps the cluster interconnection was created. At this point the cluster is ready to create resources and hierarchies to protect file systems, IP addresses, database, and SAP NetWeaver instances.

5.8 Creating resources and hierarchies to protect applications

This section describes the steps to protect resources and build hierarchies for applications.

5.8.1 Creating file system resources

First, we create file system resources needed for SAP Instances and database:

1. There are four ways to begin creating a file system resource hierarchy:
 - On the global toolbar, click on the **Create Resource Hierarchy** button as shown in Figure 5-40.

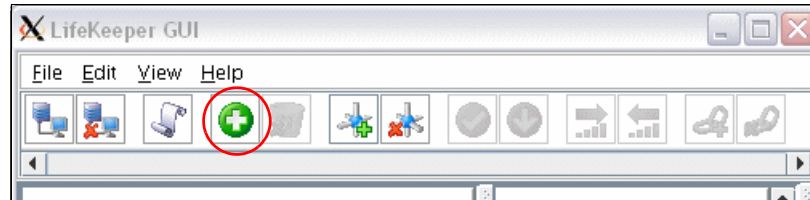


Figure 5-40 Toolbar icon Create Resource Hierarchy

- Right-click on a server icon to bring up the server context menu, then click on **Create Resource Hierarchy**
 - On the **Edit** menu, select **Server**, then click on **Create Resource Hierarchy**
 - On the server context toolbar, if displayed, click on the **Create Resource Hierarchy** button
2. A dialog entitled Create Resource Wizard is displayed with a Recovery Kit list. Select the **File System** resource for the next step as shown in Figure 5-41.

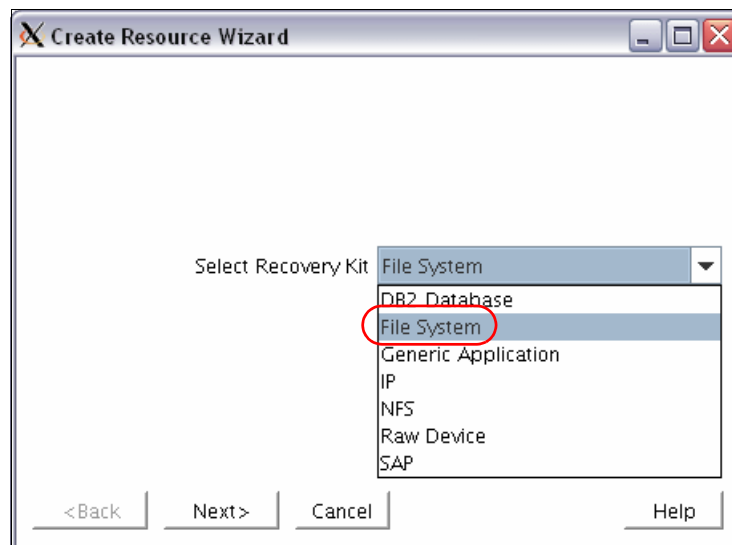


Figure 5-41 Selection of resource kit type

3. Select the Switchback Type for the new resource. The recommended Switchback Type is **intelligent**.

Note: This dictates how a resource is switched back to this server when the server comes back up after a failover. You can choose either intelligent or automatic. Intelligent switchback requires administrative intervention to switch the resource back to the primary/original server. Automatic switchback means the switchback occurs as soon as the primary server comes back on line and reestablishes LifeKeeper communication paths.

In that case, Switchback Type is set to *automatic*. The following situations are possible:

- After a successful failover of the resources because of a server failure, the users lose connection and have to reconnect to the protected application. Then the failed cluster node comes back and all the resources with Switchback Type set to automatic and all the resources it depends on switch back to the primary cluster node. This switchback does not require administrator intervention, and the users lose connectivity to the protected application again. These are two unplanned outages on the protected application.
- Another possible condition occurs when the primary cluster node crashes and the reboot cycle enters into a start/stop loop because of a persistent failure. The hierarchy including resources on which the Switchback Type is set to automatic performs failover and switchback also in a loop. The users lose the connection to the protected application each time.

In situations where the preferred Switchback Type is intelligent, the administrator is able to switch back the resources within a planned downtime.

4. Select the Server on which the resource was created first. If creation starts from the server context menu, the server is determined automatically from the server icon that was clicked on, and this step can be skipped.

5. The Create gen/filesys Resource dialog is displayed. Select the Mount Point for the file system resource hierarchy as shown in Figure 5-42.

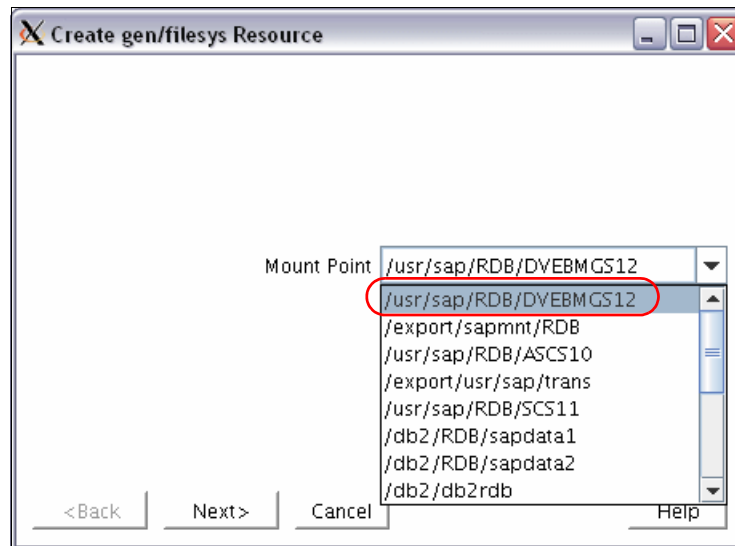


Figure 5-42 Select list for available file systems

Note: In order for a mount point to appear in the choice list, the mount point must be currently mounted. If an entry for the mount point exists in the `/etc/fstab` file, LifeKeeper removes this entry during the creation and extension of the hierarchy. When the hierarchy is deleted, an entry is placed in the `/etc/fstab` file only on the primary server.

Note: To create a File System resource that is built on a logical volume, on a Software RAID in a Device Mapper multipath configuration, then the LVM Recovery Kit, the Software RAID Recovery Kit, and the Device Mapper multipath Recovery Kit have to be installed. Otherwise, the mount point for the file system does not appear in the choice list.

6. LifeKeeper creates a default Root Tag for the file system resource hierarchy. This is the label used for this resource in the status display. Select this root tag or create your own.

- Click **Create Instance**. A window displays a message indicating the status of the instance creation as shown in Figure 5-43.

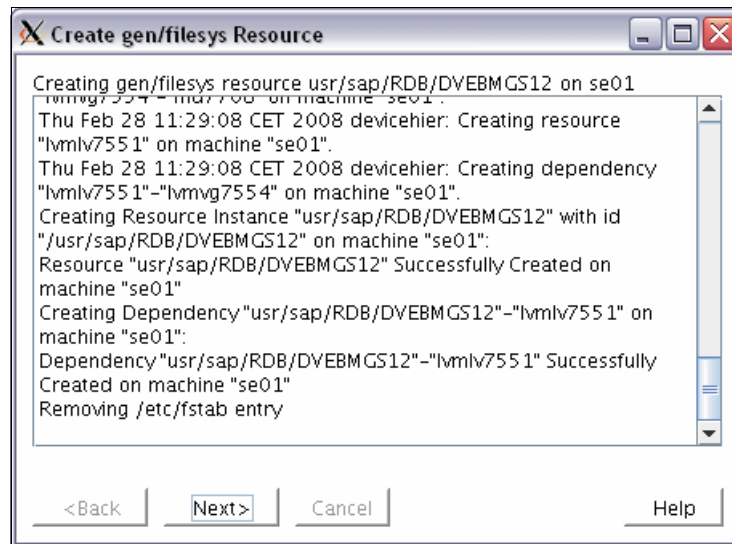


Figure 5-43 Message during creation process

- The next window displays a message that the file system hierarchy has been created successfully as shown in Figure 5-44.



Figure 5-44 Success message

9. At this point, the file system resource hierarchy has to be extended to the other node by clicking **Next**. If you click **Cancel**, a warning message says that the hierarchy exists on only one server, and it is not protected at this point.
10. Select the Switchback Type for the resource on this server. The recommended Switchback Type is **intelligent**.
11. Select a priority for the template hierarchy, relative to equivalent hierarchies on other servers. Any unused priority value from 1 to 999 is valid, where a lower number means a higher priority (the number 1 indicates the highest priority). The default value is recommended.
12. Either select or enter your own Target Priority. The default value is recommended.
13. The dialog then displays the pre-extend checks that occur next. If these tests succeed, LifeKeeper goes on to perform any steps that are required for the specific type of file system as shown in Figure 5-45.

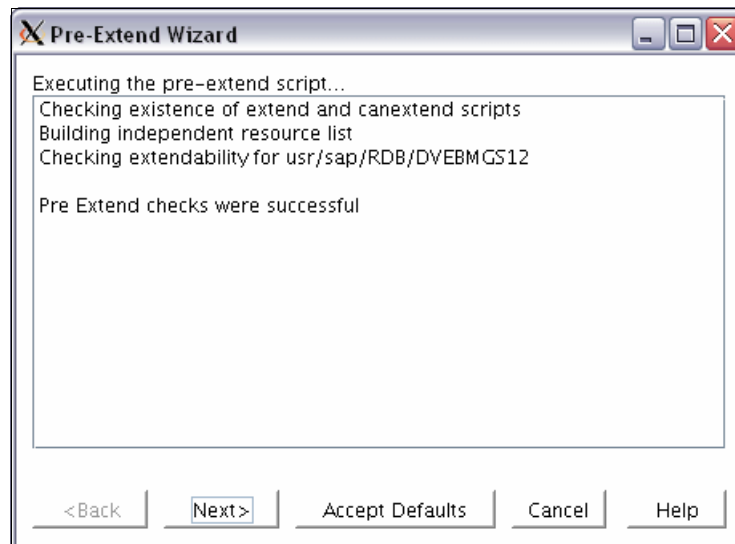


Figure 5-45 Pre extend checks

14. The Extend gen/filesys Resource Hierarchy dialog box is presented. Select the Mount Point for the file system hierarchy.
15. Select the Root Tag that LifeKeeper offers.

16. The dialog displays the status of the extend operation, which should finish with a message saying that the hierarchy has been successfully extended as shown in Figure 5-46. In a two node cluster, click **Finish** to complete this operation.

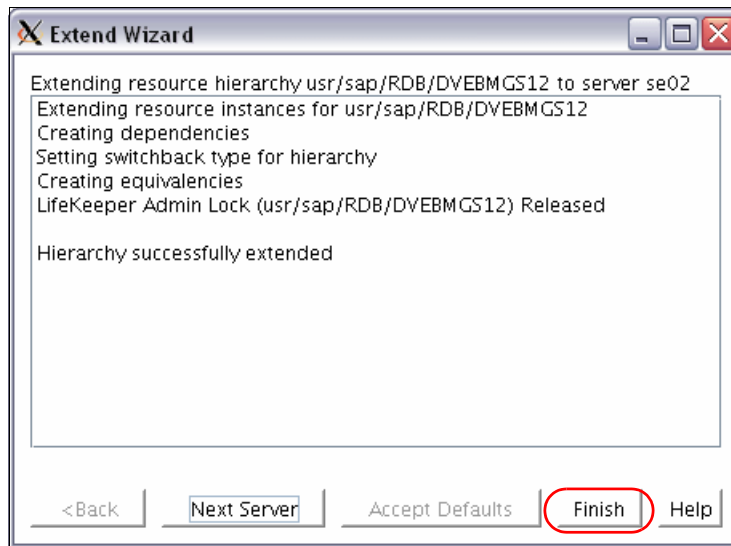


Figure 5-46 Hierarchy extended successful

Otherwise, click **Next Server** to extend the file system resource to a different server.

17. The dialog then displays verification information as the extended hierarchy is validated. When this is finished, the **Done** button is enabled. Click **Done** to finish.

Note: The **Accept Defaults** button, which is available for the Extend Resource Hierarchy option, is intended for the user who is familiar with the LifeKeeper Extend Resource Hierarchy defaults, and wants to quickly extend a LifeKeeper resource hierarchy without being prompted for input or confirmation. Users who prefer to extend a LifeKeeper resource hierarchy using the interactive, step-by-step interface of the GUI dialogs should use the **Next** button.

Repeat these steps for all file systems to protect them.

5.8.2 Creating IP resources

Secondly, we create the resources for IP addresses required for a transparent communication of users with the SAP system and for communication of the SAP system with the database:

1. There are also four ways to begin creating an IP resource, as described in “Creating file system resources” on page 166.
2. Select the IP resource from the Recovery Kit list. Click **Next**.
3. Select the Switchback Type for the new resource. The recommended Switchback Type is **intelligent**.
4. Select the Server on which the resource was created first.
5. Enter the IP resource. This is the Ip Address or symbolic name that LifeKeeper uses for this resource as shown in Figure 5-47. If a symbolic name is used, it must exist in the local `/etc/hosts` file or be accessible via a Domain Name Server (DNS). No defaults are provided for this information field.

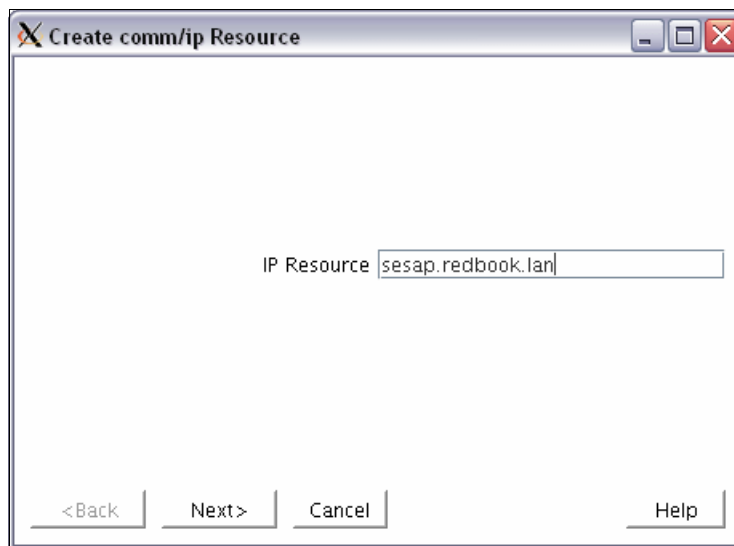


Figure 5-47 Enter the IP address

6. Select or enter the network mask, Netmask, the IP resource used on the target server as shown in Figure 5-48. Any standard Netmask for the class of the specific IP resource address is valid.

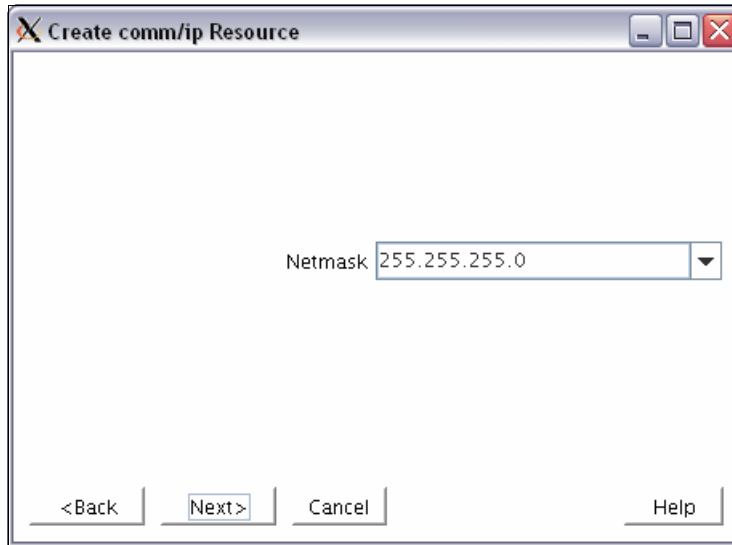


Figure 5-48 Select or enter the network mask

7. Select or enter the Network Interface where the IP resource is placed under LifeKeeper protection. This is the physical Ethernet card that the IP address is interfacing with. Valid choices depend on the existing network configuration and values chosen for the IP resource address and netmask as shown in Figure 5-49. The default value is the interface within the set of valid choices which most closely matches the address and netmask values are selected. If there are bonding interfaces configured for a higher availability, these are available to select.

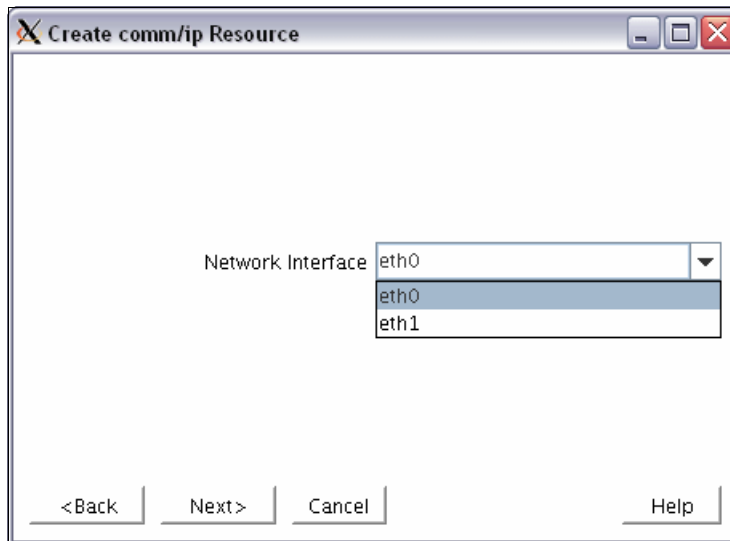


Figure 5-49 Select or enter the Network Interface

8. Select a Backup Interface to engage the IP Local Recovery feature on this server. The default value is none.

We recommend using a bonding interface to provide a higher availability of the network.

For the topics, Using the Backup Interface and Local Recovery Configuration Restrictions, refer to the *IP Recovery Kit Administration Guide*, available at:

<http://licensing.steeleye.com/documentation/linux.html>

9. Select or enter a unique IP Resource Tag name for the IP resource instance which you are creating.
10. Click **Create**. The Create Resource Wizard then creates your IP resource.
11. An information box is presented and LifeKeeper validates that all provided data is valid to create your IP resource hierarchy. If LifeKeeper detects a problem, an ERROR is displayed in the information box. If the validation is successful, the resource is created as shown in Figure 5-50.

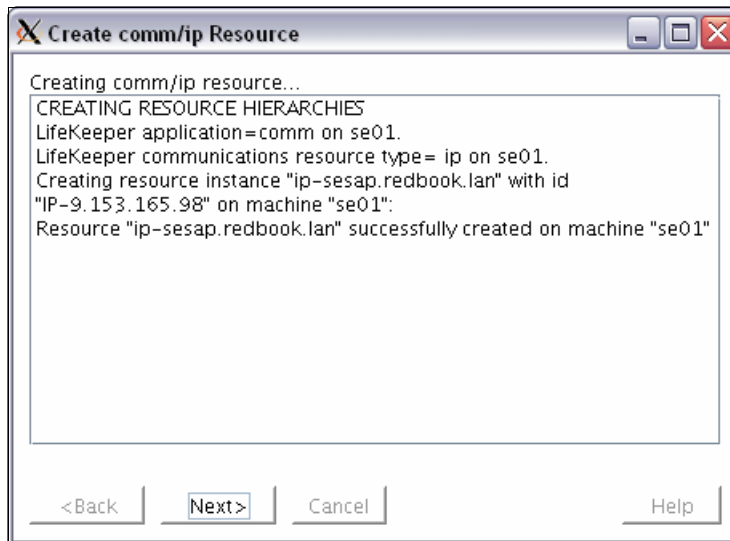


Figure 5-50 create IP resource

12. The next information box is presented, explaining that the IP resource hierarchy was created successfully, and that the IP hierarchy must be extended to another server in the cluster in order to place it under LifeKeeper protection. Click **Next**.
13. The Pre-Extend Wizard prompts you to enter the Switchback Type.
14. Select a priority for the template hierarchy. The default value is recommended.
15. Either select or enter your own Target Priority. The default value is recommended.
16. An information box is presented, explaining that LifeKeeper has successfully checked your environment and that all the requirements for extending this IP resource have been met.
17. The next two boxes are only information boxes that show the IP address and the Netmask.
18. In the next window, select the Network Interface. This is the name of the network interface that the IP resource uses on the target server.
19. Select a Backup Interface to engage the IP Local Recovery feature on this server. The default value is none.
20. Select or enter the IP Resource Tag. This is the resource tag name to be used by the IP resource being extended to the target server. Use the same tag as the one on the first server (default).

21. An information box is presented, verifying that the extension is being performed as shown in Figure 5-51. If the extension finished successfully, click **Finish** to complete this operation.

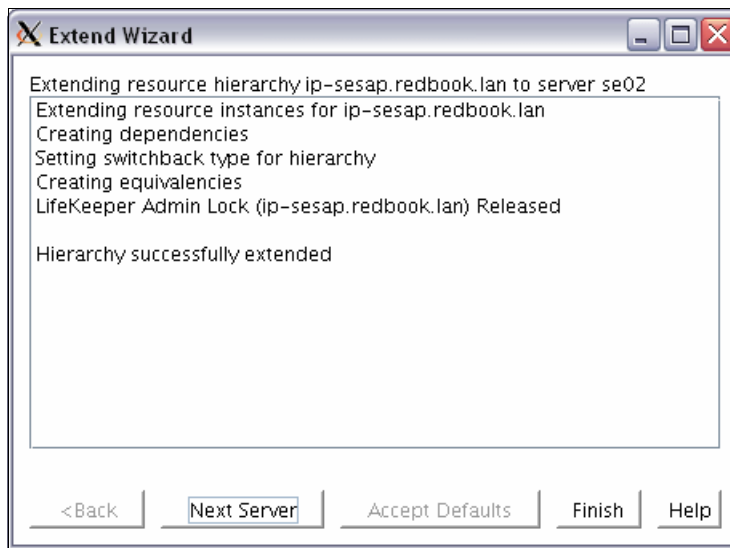


Figure 5-51 Extend IP hierarchy

Repeat these steps to create another IP resource for the database in an active/active cluster configuration.

5.8.3 Creating database resource

Thirdly, we create the resource to protect the database:

1. To start building the Database Resource, there are the same steps as described before for the File System Resources and IP Resources.
2. A dialog entitled Create Resource Wizard is presented with a Recovery Kit list. Select the Resource Kit for used database, depending on the installation (DB2, MaxDB, Oracle).
3. Select the Switchback Type for the new resource. The recommended Switchback Type is intelligent.
4. Select the Server on which the resource was first created.
5. Depending on the database used, the creation wizard shows different dialog boxes.

- Only DB2:
 - At the DB2 Instance dialog, select the name of the DB2 instance that is being protected.
 - An information box is presented, displaying information regarding the instance detected.
 - Only MaxDB:
 - The SAP DB Program Path dialog field contains by default the SAP DB Program Path found in the SAP_DBTech.ini file on the corresponding server.
 - The SAP DB Instance dialog field contains by default the name of the first SAP DB instance found on the system, for which no LifeKeeper hierarchy exists.
 - In the SAP DB System User dialog, enter the System User that owns or has permission to execute SAP DB commands. This user must exist on the corresponding server.
 - In the User_Key dialog, enter the user “c”.
 - Only Oracle:
 - Select the ORACLE_SID for the Database ID. This is the tag name that specifies the Oracle system identifier of the database being configured. An entry for this database must exist in /etc/oratab.
 - Select or enter the directory path of the ORACLE_HOME for the Database SID being protected. This is the directory location where the Oracle application is located on the primary or template server.
6. Select or enter a unique Resource Tag name for the resource instance which you are creating at the Database Tag dialog.
 7. The Create Resource Wizard then creates the database resource hierarchy. LifeKeeper validates the data entered.

During this step the Create Resource Wizard creates all dependences between the database resource and all file system resources on which the database depends.
 8. The next information box is presented, explaining that the Database resource hierarchy was created successfully, and that the Database hierarchy must be extended to another server in the cluster in order to place it under LifeKeeper protection. Click **Next**.
 9. The Pre-Extend Wizard prompts you to enter the Switchback Type.
 10. Select a priority for the template hierarchy. We recommend the default value.
 11. Either select or enter your own Target Priority. We recommend the default value.

12. An information box is presented, explaining that LifeKeeper has successfully checked your environment and that all the requirements for extending this Database resource have been met. If there are requirements that have not been met, LifeKeeper disables the **Next** button, and enables the **Back** button.
 - Click the **Back** button to make changes to the resource extension.
 - Click **Cancel** to extend the resource another time.
 - Click **Next** to launch the Extend Resource Hierarchy configuration task.
 - Click **Finish** to confirm the successful extension of Database resource instance.
13. Depending on the database used, the creation wizard shows different dialog boxes:
 - Only MaxDB:
 - By default, the SAP DB Program Path found in the SAP_DBTech.ini file on the corresponding server is shown in the SAP DB Program Path dialog box.
 - At the User_Key dialog, the user “c” entered during the creation step is shown.
 - The SAP DB Database Tag dialog shows a unique tag name for the new SAP DB database resource on the target server.
 - An information box is presented, verifying that the extension is being performed.
 - Click **Finish** to confirm the successful extension of the Database resource instance.
 - Only Oracle:
 - The next dialog box is an “information only” box displaying the ORACLE_SID tag name. It is not possible to change this designation.
 - At the next dialog box, select or enter the ORACLE_HOME for the Target Server. This is the directory location where the Oracle application is located on the backup or target server.
 - Select or enter a unique Database Tag name.
 - An information box is presented, verifying that the extension is being performed.
 - Click **Finish** to confirm the successful extension of the Database resource instance.
14. Click **Done** to exit the Extend Resources Hierarchy menu selection.

Note: Be sure to test the functionality of the new instance on both servers.

5.8.4 Creating dependences

The Database resource and dependencies between Database resource and File systems on which the database depends on have been successfully created. To comply with the Database hierarchy, the dependencies for the remaining File system resources and IP resource have to be created.

To create the dependencies, follow these steps:

1. Start the creation by right-clicking on the Database resource and select **Create dependency**.
2. A dialog entitled Create Dependency is presented, with a list of resources. Select the File system resource to create a dependency with the database resource as shown in Figure 5-52. Click **Next**.

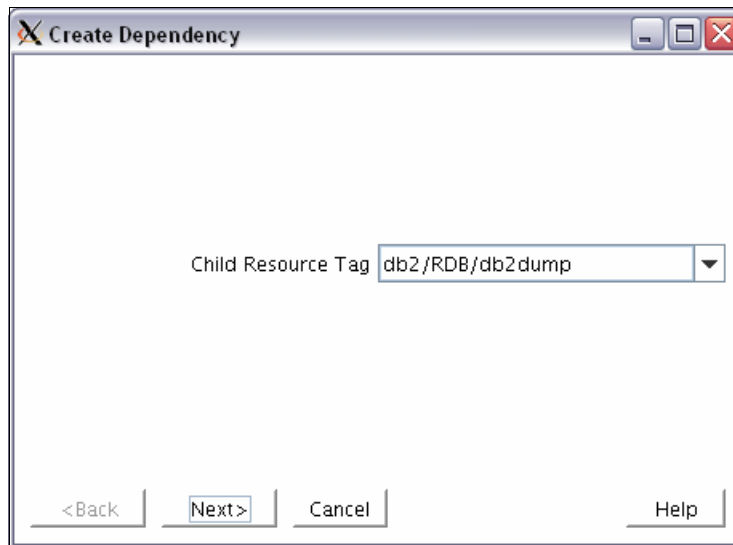


Figure 5-52 Dialog box Create Dependency

3. An information box is presented, showing the Parent and Child resource as in Figure 5-53. To create the dependency, click **Create Dependency**.

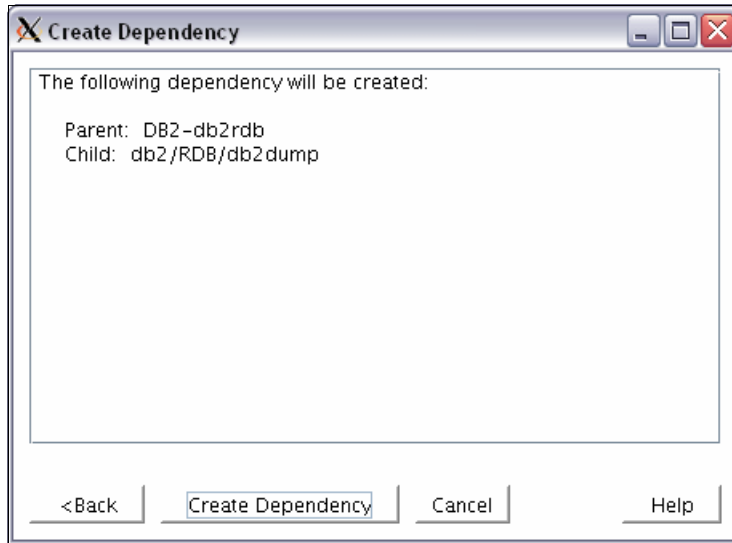


Figure 5-53 Create Dependency

4. A message box is presented, verifying that the dependency has been created successfully, as shown in Figure 5-54.

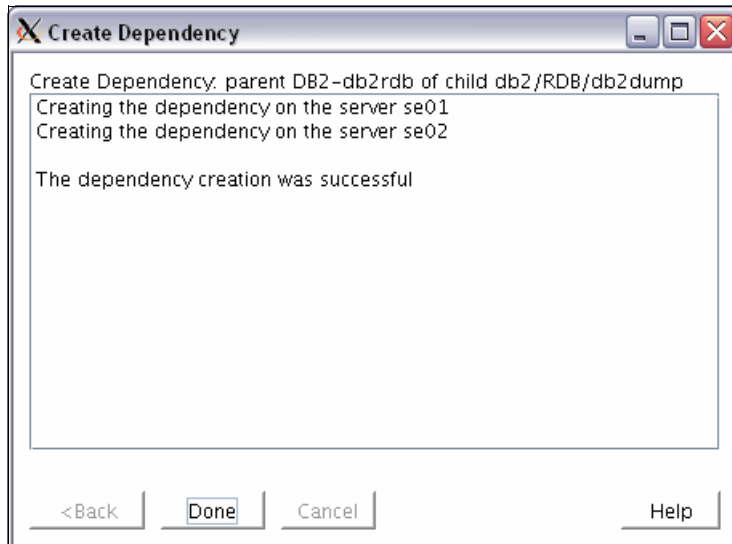


Figure 5-54 Message box

5. Click **Done** to exit from the wizard.

- Finally, the database hierarchy includes all file system resources for the database and the IP resource as shown in Figure 5-55.

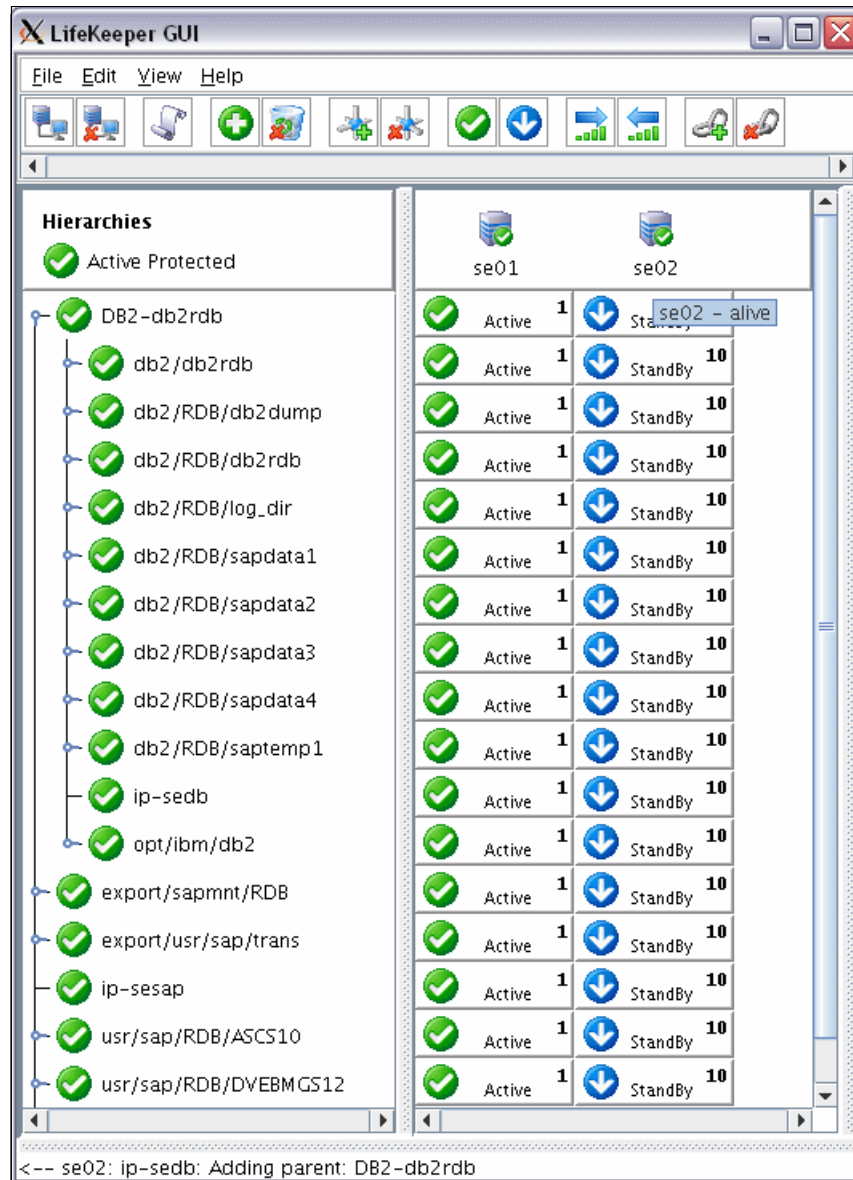


Figure 5-55 Database resource

5.8.5 Creating SAP resources

Using these steps, we create the resources required to protect SAP NetWeaver.

1. To do this, it is necessary to perform the following steps:

- Create an NFS Server resource for the exported file systems `/export/sapmnt/RDB` and `/export/usr/sap/trans` in dependency with the previously created IP resource for the SAP system.

The steps to create the NFS Server resource are the same for creating the File system resource as explained in “Creating file system resources” on page 166.

- Mount the NFS shared file directory using either the switchable IP address or the virtual SAP server name on both cluster nodes to mount point `/sapmnt/RDB`.

Insert the one entry for this mount point in `/etc/fstab` as shown in Example 5-31.

Example 5-31 Entry in /etc/fstab

```
...
sesap:/export/sapmnt/RDB /sapmnt/RDB nfs defaults,udp,intr 0 0
...
```

- Check in `/etc/services` for all SAP required entries.
- The Group IDs and User ID for SAP must be identical on both servers.
- Check the existence of all user profiles for `rdbadm`.
- Check that the soft links in the SAP directory comply with SAP standards. If they do not comply, correct the soft links for directory `/usr/sap/RDB/SYS` as shown in Example 5-32.

Example 5-32 Soft links under /usr/sap/RDB/SYS

```
se01:/usr/sap/RDB/SYS # ls -l
drwxr-xr-x 3 rdbadm sapsys 4096 2008-02-22 16:09 exe
drwxr-xr-x 3 rdbadm sapsys 4096 2008-02-22 16:09 gen
lrwxrwxrwx 1 rdbadm sapsys  18 2008-02-25 16:03 global -> /sapmnt/RDB/global
lrwxrwxrwx 1 rdbadm sapsys  19 2008-02-25 16:03 profile -> /sapmnt/RDB/profile
drwxr-xr-x 2 rdbadm sapsys 4096 2008-02-22 16:09 src
```

For directory /usr/sap/RDB/SYS/exe, see Example 5-33.

Example 5-33 Soft links under /usr/sap/RDB/SYS/exe

```
se01:/usr/sap/RDB/SYS # ls -l
lrwxrwxrwx 1 rdbadm sapsys 18 2008-02-25 16:05 uc -> /sapmnt/RDB/exe/uc
lrwxrwxrwx 1 rdbadm sapsys 24 2008-02-25 15:59 run -> /usr/sap/RDB/SYS/exe/dbg
drwxr-xr-x 2 rdbadm sapsys 4096 2008-02-22 16:09 opt
lrwxrwxrwx 1 rdbadm sapsys 19 2008-02-25 16:05 nuc -> /sapmnt/RDB/exe/nuc
lrwxrwxrwx 1 rdbadm sapsys 15 2008-02-25 16:04 dbg -> /sapmnt/RDB/exe
```

- Check the values of SAPDBHOST, j2ee/dbhost, and j2ee/dbadminurl in the DEFAULT.PFL for the virtual database server name.
2. To start building the Database Resource, follow the same steps as described in 5.8.3, “Creating database resource” on page 176.
 3. A dialog box is presented, with a drop-down list box with all recognized recovery kits installed within the cluster. Select SAP from the drop-down listing.
 4. Select the Switchback Type.
 5. Select the Server on which the resource was first created.
 6. Select the SAP SID. This is the system identifier of the SAP Add-in CI, ASCS, or SCS system being protected.
 7. Select the SAP Instance ID for the SID being protected.

Note: In ABAP+Java AddIn environments, the Java Instance is automatically detected and protected in addition to the instance chosen in the dialog.

8. This window is only presented in SAP NetWeaver ABAP Only and ABAP Add-In environments. Select the Replicated Enqueue profile if running a Replicated Enqueue in the environment. Otherwise, select **None**.
9. This window is only presented in SAP NetWeaver ABAP Add-In and Java Only environments. Select the Replicated Enqueue profile if running a Replicated Enqueue in the environment. Otherwise, select **None**.

Note: Alternately, it is possible to edit the file `/opt/LifeKeeper/subsys/appsuite/resources/sap/REPENQ_<TAG>`, where `<TAG>` is the tag name of the SAP resource on the local server. Each line in this file contains information about one instance. There are two fields on each line, delimited by a colon. The fields are:

- ▶ The ID of the instance
- ▶ The Replicated Enqueue profile file name

Any changes made to these files takes immediate effect. To add an instance for LifeKeeper to control on the SAP Backup Server, add a line at the end of the file on both servers. To remove an instance, simply delete the line on both servers.

10. Select all IP resources that are used to communicate with the Add-in CI. Note that IP resources that are included in previously created NFS hierarchies are also listed here. If these are not selected, they are included in the hierarchy by default anyway.
11. An information box is presented, and LifeKeeper validates that all provided data is valid to create an SAP resource hierarchy. If LifeKeeper detects a problem, an ERROR is shown in the information box. If the validation is successful, your resource is created. There can also be errors or messages output from the SAP startup scripts that are displayed in the information box.
12. The next information box is presented, explaining that the SAP resource hierarchy is successfully created. The hierarchy must be Extend to another server in the cluster in order to place it under LifeKeeper protection.
13. The Pre-Extend Wizard prompts you to enter the Switchback Type.
14. Select a priority for the template hierarchy. The default value is recommended.
15. Either select or enter your own Target Priority. The default value is recommended.
16. An information box is presented, explaining that LifeKeeper has successfully checked the environment and that all the requirements for extending this SAP resource have been met. If there were some requirements that had not been met, LifeKeeper would not allow you to select the **Next** button, and the **Back** button would be enabled.
17. If there is an SAP AS instance on the Target Server that should be running when the Add-in CI, ASCS, or SCS is in-service on the Template Server, but shut down when the Add-in CI, ASCS, or SCS is in-service on the Target Server, the SID for that instance should be selected from the list. Alternatively, select **None**.
 - If a SID was selected, select the SAP Instance for the selected SID.

18. Select or enter the SAP Tag. The default Tag is identical to the Tag on the Template server.
19. An information box is presented, verifying that the extension is being performed.
20. If you click **Finish**, another dialog box is presented, confirming that LifeKeeper has successfully extended your SAP resource.

The current states of the hierarchies are shown in Figure 5-56.

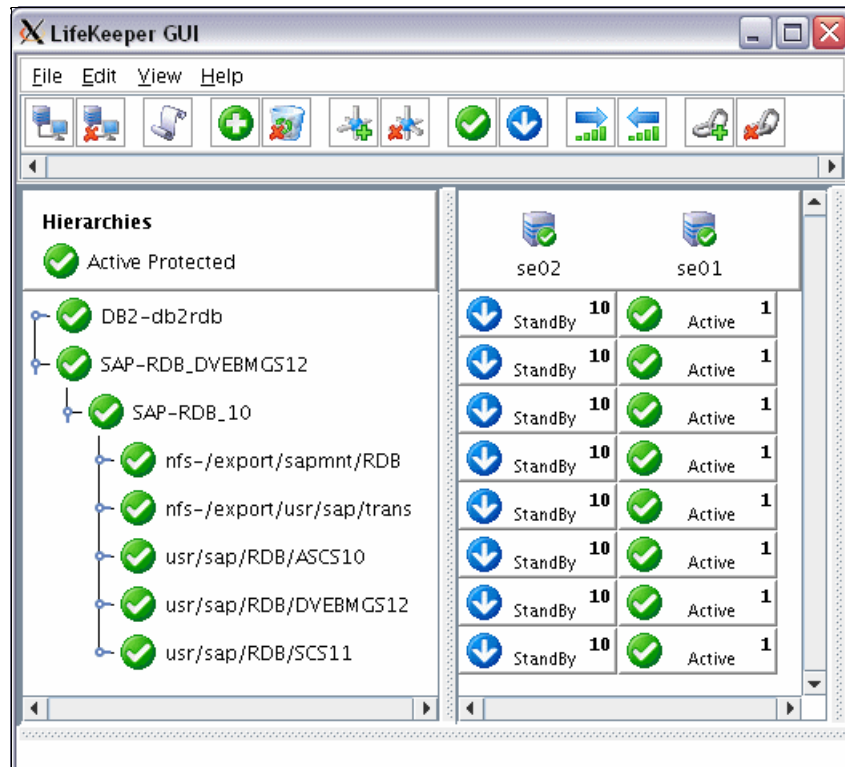


Figure 5-56 SAP hierarchy

At this point the ABAP Central Services (ASCS) and the Java Central Services (SCS) of the SAP NetWeaver System are protected. These are the SPOFs of the SAP NetWeaver system.

In a normal installation, a central instance is also installed on the server. This instance is at this time not protected by LifeKeeper. To include this instance in LifeKeeper protection, an additional resource has to be created. This resource is a generic application type. How to create this resource is explained in the next section.

Creating Resource for the Central Instance

To create a resource of type Generic Application, the scripts to control and check the application have to be created as autonomous.

A resource of type generic application needs a restore script to start the application, a remove script to stop the application, and optionally a quickCheck script to check and a recovery script to recover the application.

The LifeKeeper installation includes some template scripts to control a generic application. These scripts are located in the directory `/opt/LifeKeeper/lkadm/subsys/gen/app/templates`. It is possible to use Shell scripts or Perl scripts to create resources for generic applications.

The following listing shows adapted template scripts to include a central SAP instance in a LifeKeeper configuration. At the point `# Add your code here`, the custom source code must be inserted.

Example 5-34 shows the template for the restore script:

Example 5-34 Template for restore script

```
...
# Add your code here to check the status of your application.
# If it is already running and responding properly then exit 0 now.
# Note: If quickCheck was implemented then a similar form of check should
# be performed at the end of this script to determine if the
# application is functioning properly.
# Set variables
    INSTANCE=~$LKROOT/bin/getinfo $TAG | awk '{ print $1 }'~
    SIDADM=~$LKROOT/bin/getinfo $TAG | awk '{ print $2 }'~
    EXECPATH=~$LKROOT/bin/getinfo $TAG | awk '{ print $3 }'~
    SAPHOST=~$LKROOT/bin/getinfo $TAG | awk '{ print $4 }'~

# Check
    ps -ewwf | grep $(echo $TAG | sed "s/SAP-//g") | grep $SIDADM | grep
-v grep
    health=$?;
    if [ $health -eq 0 ]; then
        pl "LifeKeeper: restore successful for $TAG\n";
        err=0
        exit 0;
    fi

#

# If your application was not already responding properly, then add
# code here to start your application
# Start
```

```

        su - $SIDADM -c "$EXECPATH/startsap $INSTANCE $SAPHOST"
        return_code=$?
        if [ $return_code -eq 1 ]; then #start failed
            pl "LifeKeeper: restore failed for $TAG\n";
            exit 1
        fi
#
# Now check to ensure that the application stopped, else exit 1.
# Check
        ps -ewwf | grep $(echo $TAG | sed "s/SAP-//g") | grep $SIDADM | grep
-v grep
        health=$?;
        if [ $health -ne 0 ]; then
            pl "LifeKeeper: restore failed for $TAG\n";
            exit 1;
        fi

pl "LifeKeeper: restore successful for $TAG\n";
err=0
exit 0

```

Example 5-35 shows the template for the remove script:

Example 5-35 Template for remove script

```

...
# Add your code here to check the status of your application.
# If it is already stopped properly then exit 0 now.
# Set variables
        INSTANCE=~$LKROOT/bin/getinfo $TAG | awk '{ print $1 }'`
        SIDADM=~$LKROOT/bin/getinfo $TAG | awk '{ print $2 }'`
        EXECPATH=~$LKROOT/bin/getinfo $TAG | awk '{ print $3 }'`
        SAPHOST=~$LKROOT/bin/getinfo $TAG | awk '{ print $4 }'`

# Check
ps -ewwf | grep $(echo $TAG | sed "s/SAP-//g") | grep $SIDADM | grep -v grep
        health=$?;
        if [ $health -ne 0 ]; then
            pl "LifeKeeper: remove successful for $TAG\n";
            err=0
            exit 0;
        fi
#
# If your application was not already stopped properly, then add

```

```

# code here to stop your application
# Stop
        su - $SIDADM -c "$EXECPTH/stopsap $INSTANCE $SAPHOST"
        return_code=$?
        if [ $return_code -eq 1 ]; then          #stop failed
            pl "LifeKeeper: remove failed for $TAG\n";
            err=1
            exit 1
        fi
#
# Now check to ensure that the application stopped, else exit 1.
# Check
        ps -ewwf | grep $(echo $TAG | sed "s/SAP-//g") | grep $SIDADM | grep
-v grep
        health=$?;
        if [ $health -eq 0 ]; then
            pl "LifeKeeper: remove failed for $TAG\n";
            exit 1;
        fi

pl "LifeKeeper: remove successful for $TAG\n";
err=0
exit 0

```

Example 5-36 shows the template for the quickCheck script:

Example 5-36 Template for quickCheck script

```

...
# Add your code here to check the state of the application
# Checks should verify application accessibility and
# sanity (eg. processes active & responding )
# It is important that this script terminate after a
# short period of time, so that monitoring of other
# resources can continue.
# Set variables
        INSTANCE=~$LKROOT/bin/getinfo $TAG | awk '{ print $1 }'~
        SIDADM=~$LKROOT/bin/getinfo $TAG | awk '{ print $2 }'~
        EXECPTH=~$LKROOT/bin/getinfo $TAG | awk '{ print $3 }'~
        SAPHOST=~$LKROOT/bin/getinfo $TAG | awk '{ print $4 }'~

# Check
        ps -ewwf | grep $(echo $TAG | sed "s/SAP-//g") | grep $SIDADM | grep
-v grep

```

```

health=$?;

#           ps -ewwf | grep "<my_process>"
#           $health=$?;
#####
# Exit with appropriate exit code and call sendevent if necessary
#####

# For success
[ $health -eq 0 ] && exit 0;

# For failure
exit 1;

```

Example 5-37 shows the template for the recovery script:

Example 5-37 Template for recovery script

```

...
# Add your code here to check that application is still failed
# Note: Checks should verify application accessibility and
#       sanity (eg. processes active & responding )
# Check
        INSTANCE=~$LKROOT/bin/getinfo $TAG | awk '{ print $1 }'~
        SIDADM=~$LKROOT/bin/getinfo $TAG | awk '{ print $2 }'~
        EXECPATH=~$LKROOT/bin/getinfo $TAG | awk '{ print $3 }'~
        SAPHOST=~$LKROOT/bin/getinfo $TAG | awk '{ print $4 }'~
        ps -ewwf | grep $(echo $TAG | sed "s/SAP-//g") | grep $SIDADM | grep
-v grep
        health=$?;
        if [ $health -eq 0 ]; then
                pl "LifeKeeper: restore successful for $TAG\n";
                err=0
                exit 0;
        fi

# Add your code here to attempt to restart your application
# This may require code for stopping and restarting your failed
# application
# Stop
        su - $SIDADM -c "$EXECPATH/stopsap $INSTANCE $SAHOST"
        return_code=$?
        if [ $return_code -eq 1 ]; then                #stop failed
                pl "LifeKeeper: remove failed for $TAG\n";

```

```

        err=1
        exit 1
    fi

# Start
    su - $SIDADM -c "$EXECPTH/startsap $INSTANCE $SAPHOST"
    return_code=$?
    if [ $return_code -eq 1 ]; then #start failed
        pl "LifeKeeper: restore failed for $TAG\n";
        exit 1
    fi

#

# Now repeat the health check to ensure that the application started
# correctly, else exit 1.
# Check
    ps -ewwf | grep $(echo $TAG | sed "s/SAP-//g") | grep $SIDADM | grep
-v grep
    health=$?;
    if [ $health -ne 0 ]; then
        pl "LifeKeeper: restore failed for $TAG\n";
        exit 1;
    fi

pl "LifeKeeper: $CMD: Local Recovery successful for $TAG\n";
err=0
exit 0

```

After a test of operability of these scripts, the resource can be created:

1. To start building the Generic Application resource, there are the same steps as described before in this section for the other resources.
2. Select the Recovery Kit Type Generic Application from the list box.
3. Select the Switchback Type. The default selection is **intelligent**.
4. In the next box, type the location of the adapted Restore script for the new resource.

Note: The scripts used for this resource must be executable by user root.

5. Next, type the location for the adapted Remove Script.

6. To perform a monitoring of the resource, type the location of the adapted quickCheck Script.
7. For the possibility to restart the resource in a case of failure, type the location of the adapted Recovery Script.
8. In case you want to use the sample scripts shown previously, at the next box in the text field for Application Info, type the values for Instance name, SAP operating system user, executable directory for SAP, and the virtual host name for the SAP instance separated by blanks as shown in Figure 5-57.

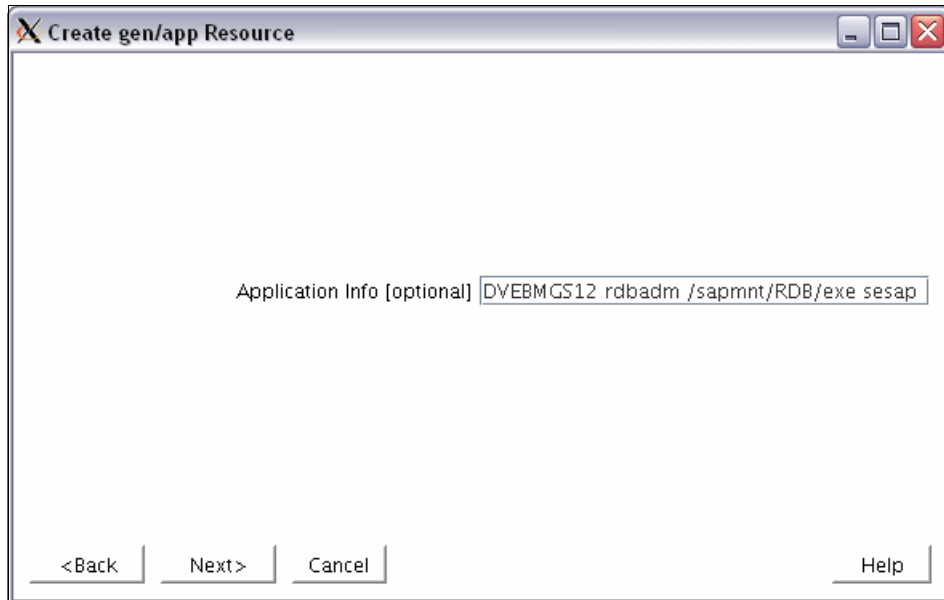


Figure 5-57 Application Info Field

9. In the next dialog box, select **Yes** for Bring Resource In Service.
10. Select a Tag for the new resource by using the following rule. The Tag begins with "SAP-", followed by the <SAP SID>_ and <SAP Instance name>. Figure 5-58 shows an example for an SAP System named RDB and an Instance DVEBMGS12.

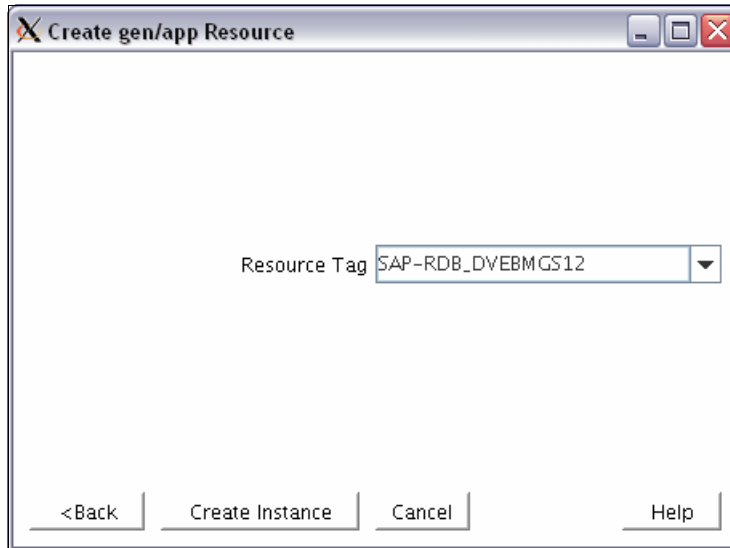


Figure 5-58 Resource Tag

11. An information box is presented, and LifeKeeper validates that all provided data is valid to create a generic resource. If the validation is successful, your resource is created.
12. The next information box is presented, explaining that the resource is successfully created. The hierarchy must be extended to another server in the cluster in order to place it under LifeKeeper protection.
13. Select the Switchback Type. Default selection is **intelligent**.
14. Select a priority for the template hierarchy. The default value is recommended.
15. Either select or enter your own Target Priority. The default value is recommended.
16. An information box is presented, explaining that LifeKeeper has successfully checked the environment and that all the requirements for extending this resource have been met.
17. Select a Tag for the new resource by using the rule described before ("SAP-"<SAP SID>_<SAP Instance name>).
18. Type the same values in the Application Info Field described before (Instance name, SAP operating system user, executable directory for SAP and the virtual host name for the SAP instance separated by blanks).
19. An information box is presented, verifying that the extension is being performed.

20. If you click **Finish**, another dialog box is presented, confirming that LifeKeeper has successfully extended your Generic Resource.
 21. Create a dependency between the new created Generic Resource for the SAP Central Instance as parent resource and resource hierarchy for the protected SAP Central Services. The application flow is the same as described before in 5.8.4, "Creating dependences" on page 179.
- Figure 5-59 shows the SAP NetWeaver hierarchy, including Central instance and SAP central services.

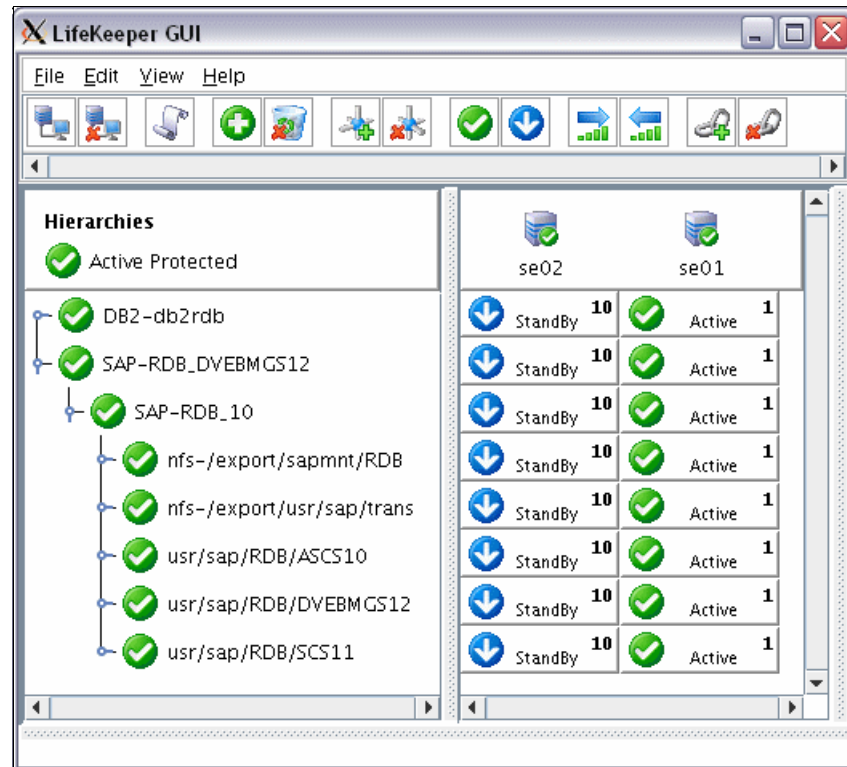


Figure 5-59 SAP NetWeaver hierarchy including Central instance and SAP central services

For an active/active configuration in which by default the database hierarchy is active on one cluster node and the database hierarchy is active on the other cluster node, change the priority of one hierarchy.

In case of an active/passive configuration in which the SAP hierarchy and the database hierarchy are active at the same time on one cluster node, create a dependence between the resource for the SAP Central Services as parent resource and the resource of the database as child resource, for example, SAP-RDB_10 and DB2-db2rdb.

Changing the priority

To change the priority of one hierarchy, for example, for the database hierarchy, follow these steps:

1. Right-click on the database resource icon, then click **Properties** when the resource context menu is presented, as shown in Figure 5-60.

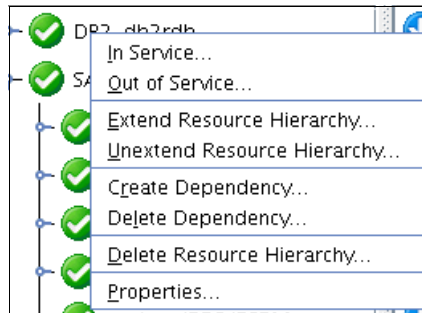


Figure 5-60 Resource context

2. In the Equivalencies Tab of the Properties window, select the cluster node with current priority 1 and click **Down** and **Apply**. The priorities of all resources included in this hierarchy changed to the new priority order as shown in Figure 5-61.

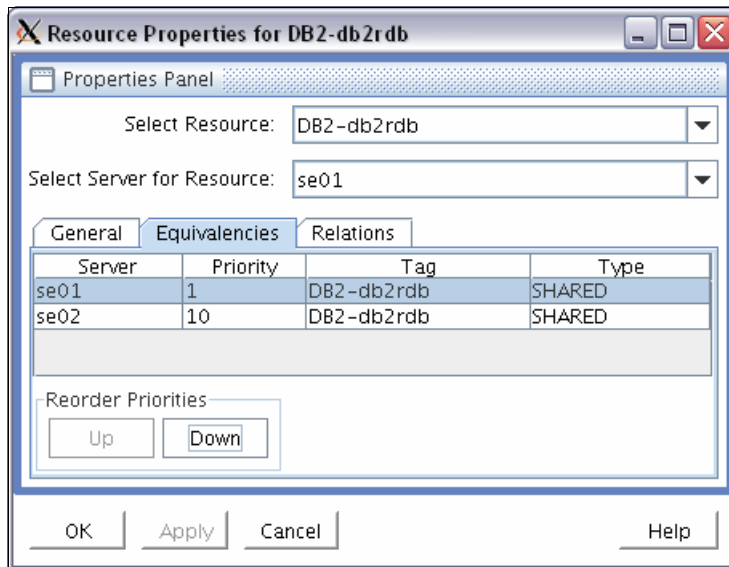


Figure 5-61 Properties window

3. A message box is presented, giving the state of the process and notifying you of the successful ending, as shown in Figure 5-62.

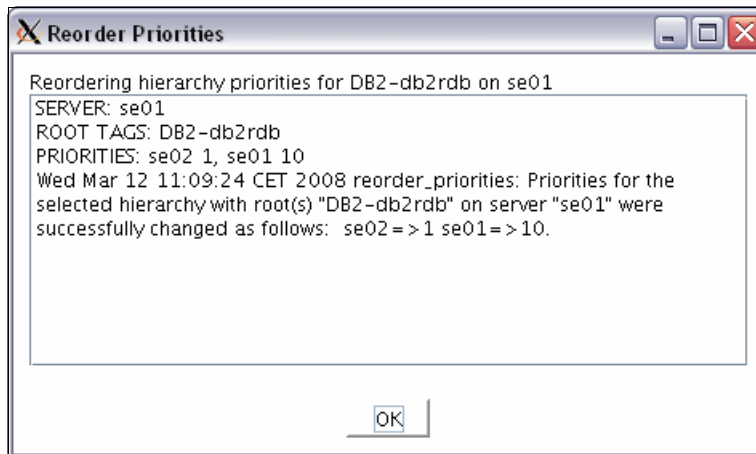


Figure 5-62 Message box

4. Now it is possible to distribute the hierarchies according the priorities. Figure 5-63 shows the final hierarchy configuration and distribution of SAP NetWeaver, and the DB2 database.

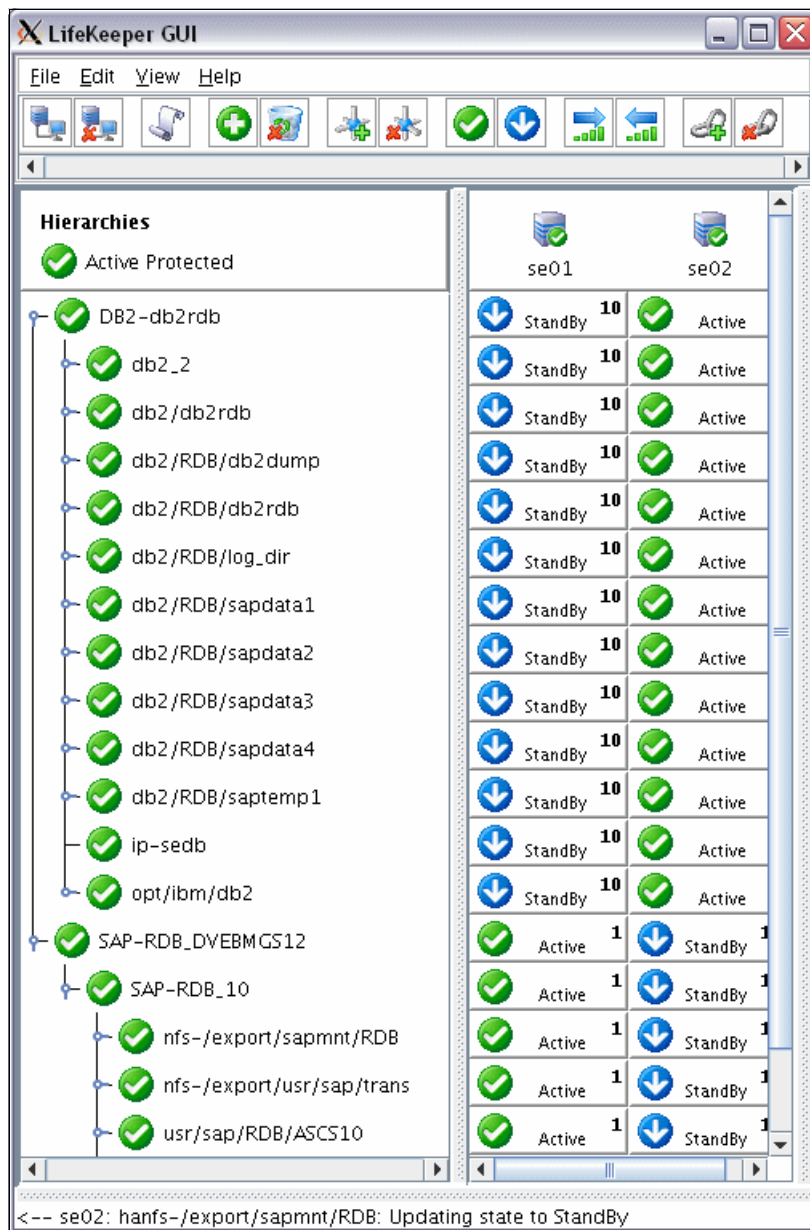


Figure 5-63 SAP NetWeaver and DB2 database in an active/active configuration

At this point the installation and configuration tasks for LifeKeeper are complete. In Chapter 6., “Testing and failover scenarios” on page 197, the functionality of the LifeKeeper is tested.



Testing and failover scenarios

In this chapter we describe each of the failover scenarios tested. Each test is presented in a step by step approach with outcome results highlighted.

We cover the following topics:

- ▶ 6.1, “Test methodology”
- ▶ 6.2, “Failover scenarios”

6.1 Test methodology

For each of the defined tests, a set of common steps were performed. This section describes these steps and the methodology that was followed:

6.1.1 Test steps

Testing was performed in three major steps:

- 1. Status verification
- 2. Component failure
- 3. Behavior of the system

Status verification

Prior to any test, a step was completed to verify the current status of the system. The built-in commands and scripts were executed to ensure that the system or component was working properly. Availability of file systems, central services, enqueue, and Java instances are each verified.

- The availability of the file systems in the active host is shown in Example 6-1.

Example 6-1 Mounted file systems in the active server

```
se01:~ # df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/systemvg-rootlv	508M	260M	249M	52%	/
udev	4.0G	328K	4.0G	1%	/dev
/dev/sda1	130M	8.2M	115M	7%	/boot
/dev/mapper/systemvg-home1v	252M	4.7M	247M	2%	/home
/dev/mapper/systemvg-optlv	1.5G	1.4G	161M	90%	/opt
/dev/mapper/systemvg-tmplv	508M	136M	373M	27%	/tmp
/dev/mapper/systemvg-usrlv	2.0G	1.9G	93M	96%	/usr
/dev/mapper/systemvg-varlv	508M	115M	393M	23%	/var
/dev/mapper/systemvg-usrsaplv	496M	18M	453M	4%	/usr/sap
/dev/mapper/systemvg-usrsaprd1v	3.0G	1.8G	1.2G	61%	/usr/sap/RDB

shm	17G	24M	17G	1%	/dev/shm
/dev/mapper/sedbv-g-RDB_sapdata1lv	30G	11G	18G	38%	/db2/RDB/sapdata1
/dev/mapper/sedbv-g-RDB_sapdata2lv	9.9G	7.0G	2.4G	75%	/db2/RDB/sapdata2
/dev/mapper/sedbv-g-db2rdb1lv	1.8G	81M	1.6G	5%	/db2/db2rdb
/dev/mapper/sedbv-g-RDB_saptemp1lv	1008M	33M	925M	4%	/db2/RDB/saptemp1
/dev/mapper/sedbv-g-RDB_sapdata4lv	9.9G	6.2G	3.3G	66%	/db2/RDB/sapdata4
/dev/mapper/sedbv-g-db2bin1lv	1008M	786M	172M	83%	/opt/ibm/db2
/dev/mapper/sedbv-g-RDB_log_dir1lv	1.5G	681M	755M	48%	/db2/RDB/log_dir
/dev/mapper/sedbv-g-RDB_sapdata3lv	9.9G	7.0G	2.4G	75%	/db2/RDB/sapdata3
/dev/mapper/sedbv-g-RDB_db2dump1lv	124M	35M	84M	30%	/db2/RDB/db2dump
/dev/mapper/sedbv-g-RDB_db2rdb1lv	248M	28M	208M	12%	/db2/RDB/db2rdb
/dev/mapper/sedbv-g-db2_2lv	25G	5.3G	19G	23%	/db2_2
/dev/mapper/sesapvg-ascs10lv	496M	115M	356M	25%	/usr/sap/RDB/ASCS10
/dev/mapper/sesapvg-scs11lv	1008M	131M	826M	14%	/usr/sap/RDB/SCS11
/dev/mapper/sesapvg-exportsapmntlv	2.0G	601M	1.3G	32%	/export/sapmnt/RDB
/dev/mapper/sesapvg-usrsaptranslv	496M	17M	455M	4%	/export/usr/sap/trans
/dev/mapper/sesapvg-dvebmsg12lv	4.0G	2.9G	940M	76%	/usr/sap/RDB/DVEBMGS12
sesap:/export/sapmnt/RDB	2.0G	601M	1.3G	32%	/sapmnt/RDB

- The availability of the ABAP Central Services is shown in Example 6-2.

Example 6-2 Availability of the ABAP Central Services ASCS10

```
se01:~ # telnet sesap 3310
Trying 9.153.165.98...
Connected to sesap.
Escape character is '^['.
```

- The availability of the Java Central Services is shown in Example 6-3.

Example 6-3 Availability of the Java Central Services SCS11

```
se01:~ # telnet sesap 3311
Trying 9.153.165.98...
Connected to sesap.
Escape character is '^['.
```

- The availability of the ABAP system through the SAP client is shown in Figure 6-1.

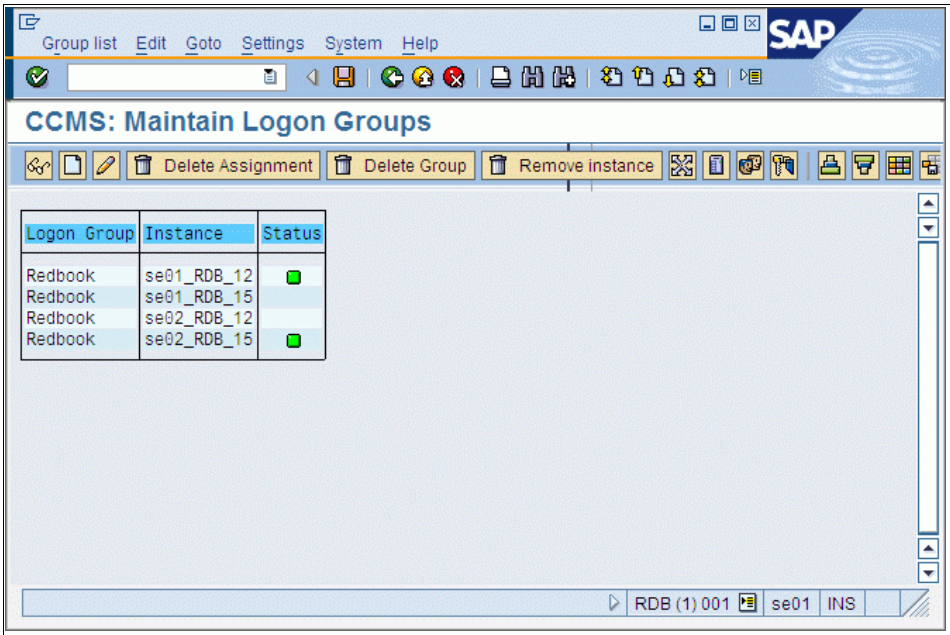


Figure 6-1 ABAP SAP system RDB through the logon group

- The availability of the Java instance through the Web server is shown in Figure 6-2

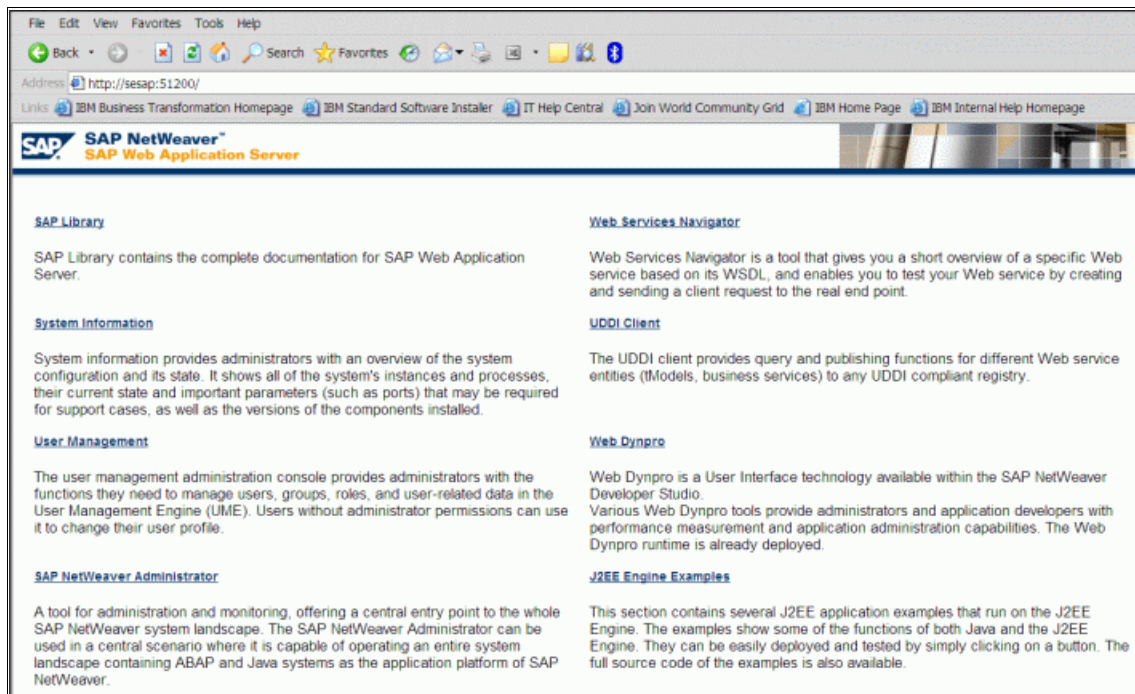


Figure 6-2 Checking availability of Java instance

- The status of the ABAP replication server is performed by running this command with the SAP system administration account (menu option):

ensmon -H <hostname of central service> -I <instance number> 2

This check is shown in Example 6-4.

Example 6-4 Checking enqueue replication status

```
se01:rdbadm 57> ensmon -H sesap -I 10 2
```

```

Try to connect to host sesap service sapdp10
get replinfo request executed successfully

Replication is enabled in server, repl. server is connected
Replication is active
=====

Information from the enqueue server side:
=====

general information (configuration):
-----
network protocol name      : no name yet
network protocol version   : 0
replication protocol version : 3
select timeout on enq. side : 0ms
keepalive timeout on enq. side: 0ms
network fragment size      : 0 Bytes
replication fragment size   : 0 Bytes
replication store size      : 1000 entries
replication queue size      : 1000 entries

```

Figure 6-3 ABAP enqueue server replication status

- The status of the Java replication server is shown in Figure 6-4.

```

Try to connect to host sesap service sapdp11
get replinfo request executed successfully

Replication is enabled in server, repl. server is connected
Replication is active
=====

Information from the enqueue server side:
=====

general information (configuration):
-----
network protocol name      : no name yet
network protocol version   : 0
replication protocol version : 3
select timeout on enq. side : 0ms
keepalive timeout on enq. side: 0ms
network fragment size      : 0 Bytes
replication fragment size   : 0 Bytes
replication store size      : 1000 entries
replication queue size      : 1000 entries

```

Figure 6-4 Java enqueue server replication status

Component failure

After the status of each component was checked, a failure was provoked with an abrupt stop of the component by issuing the **kill -9** command to terminate the process, or by simulating a crash with the command **reboot -nf**, which performs an immediate restart of the server without shutting down any application and without synchronizing the mounted file system.

Behavior of the system

In this step, the actual behavior of the system after the failure is compared with the expected one. For every tested scenario, we provide a description of the expected behavior of the tested component.

A workload was generated in the system performing administrative tasks such as client copy and generation of programs.

6.2 Failover scenarios

The scenarios tested within this book cover mainly unplanned outages resulting from a failure of a single component or the entire server. Table 6-1 describes each of the defined test scenarios:

Table 6-1 Failure scenarios

Component Failure Scenario	Description
Active server failure	Failure of the server where the central services and the database are running
Standby server failure	In the test scenario, failure of the server where the application server and the replication servers are running
ABAP central services failure	Failure of the ABAP enqueue server, or the message server
Java central services failure	Failure of the Java enqueue server, or the message server
Central instance failure	Failure of the whole set of central instance components. Failure of individuals components are automatically restarted by the SAP system
Application server failure	Failure of the whole set of central instance components. Failure of individuals components are automatically restarted by the SAP system
Database failure	Failure of the database system
Replication server failure	Failure of the replication server in the standby server
NFS Server Failure	Failure of the NFS server

6.2.1 Failure of the active server

In this scenario, the failure of the server se01 was tested. All the resources were running in the se01 server, except for the additional application instance and the enqueue replication servers.

Table 6-2 summarizes the execution of the test.

Table 6-2 Status of the active server before the failure

Purpose	Simulate a failure of the active server.
Status before the test	Servers se01 and se02 were up and running: <ul style="list-style-type: none">▶ DVEBMS12 central instance was running in se01.▶ ASCS10 ABAP instance was running in se01.▶ SCS11 Java instance was running in se01.▶ RDB database was running in se01.▶ D15 application instance was running in se02.▶ ERS28 and ERS29 enqueue replication instances were running in se02.
Execution steps	Reboot the server se01.
Expected results	LifeKeeper for Linux would have detected the failure and switched resources to the se02 server automatically.
Status check	Verification of the status of all components.
Behavior	LifeKeeper for Linux detected the failure in server se01 and automatically switched the resources for the se02 server successfully.

In order to verify the behavior of the SAP system RDB lock table, a mass test locking was generated with transaction SM12. The lock list is shown in Figure 6-5.

Connections for both SAP instances in se01 and se02 servers were opened from SAPGUI client and Web browser.

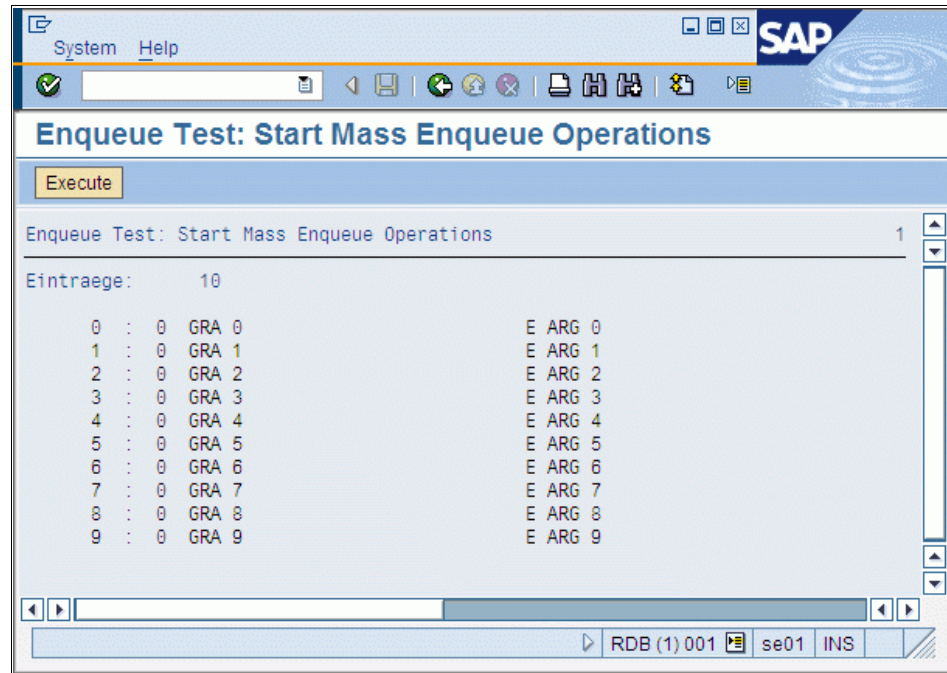


Figure 6-5 Lock entries in SAP enqueue table

After the simulation of a crash using the **reboot** command, LifeKeeper for Linux detected the failure and automatically switched the resources to the standby server. The status of the system was as follows:

- ▶ Sessions connected in se01 (DVEBMSG12) crashed as expected.
- ▶ Sessions in se02 (D15) were still running.
- ▶ Locks in the SAP system were not reset.

Figure 6-6 shows the active servers both running on server se02.

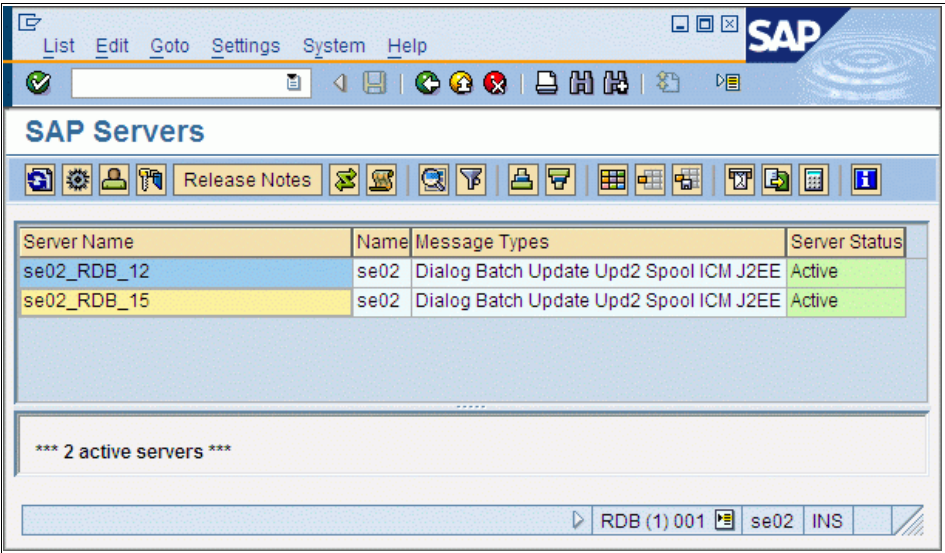


Figure 6-6 Active instances in se02

6.2.2 Failure of the standby server

The standby server (se02) in the test environment was running the SAP application server D15 and the enqueue replications servers (ERS28 and ERS29). A failure was simulated in the se02 server to check how the applications running on the se01 server would behave. See Table 6-3.

Table 6-3 Summary of the standby server failure simulation

Purpose	Simulate a failure of the standby server.
Status before the test	<div>Servers se01 and se02 were up and running:</div> <div><div>▶ DVEBMS12 central instance was running in se01.</div><div>▶ ASCS10 ABAP instance was running in se01.</div><div>▶ SCS11 Java instance was running in se01.</div><div>▶ RDB database was running in se01.</div><div>▶ D15 application instance was running in se02.</div><div>▶ ERS28 and ERS29 enqueue replication instances were running in se02.</div></div>
Execution steps	Simulate a crash with <code>reboot -nf</code> command.
Expected results	LifeKeeper for Linux would have detected the failure and not taken action until the communication with the standby server is reestablished.

Purpose	Simulate a failure of the standby server.
Status check	Verification of the status of all components.
Behavior	LifeKeeper for Linux detected the failure of the standby server and restarted the instances in se02 when the server was ready.

Four SAP sessions were connected into the SAP system RDB as follows:

- ▶ ABAP session on se01 instance DVEBMGS12 using the SAPGUI client
- ▶ Java session on se01 instance DVEBMGS12 using a Web browser on the address <http://sesap:51200>
- ▶ ABAP session on se02 instance D15 using the SAPGUI client
- ▶ Java session on se02 instance D15 using a Web browser on the address <http://se02:51500>

The Web page to the D15 instance in se02 is shown in Figure 6-7.

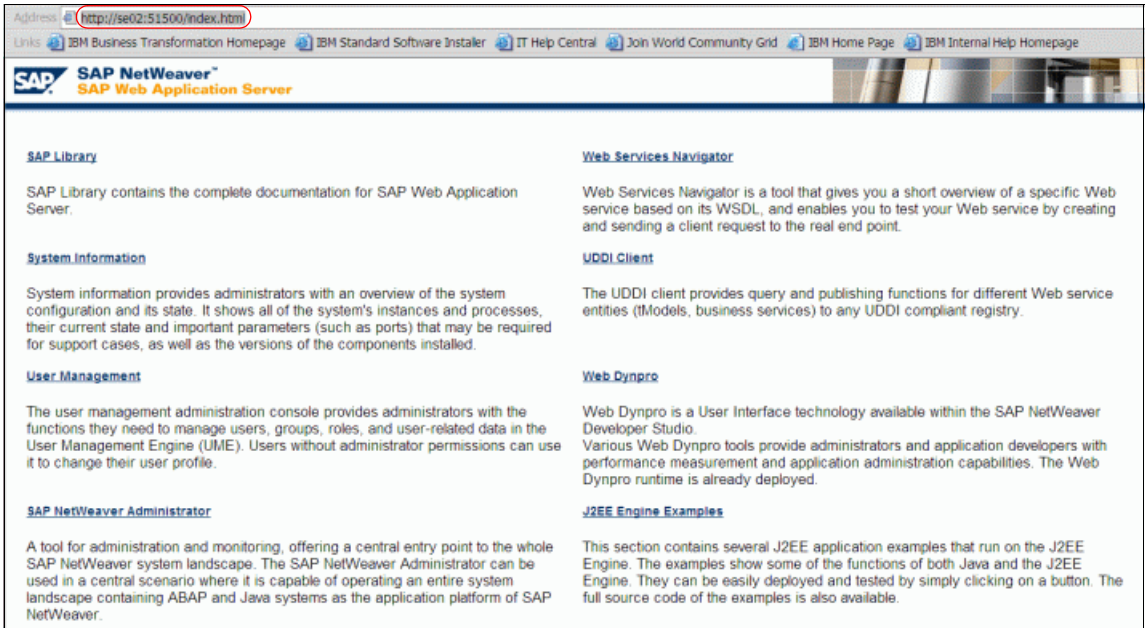


Figure 6-7 Connection to the D15 instance on se02

The crash was simulated in the se02 server with the command **reboot-nf**, which performs an immediately restart of the server without shutting down applications and without synchronizing the file system.

The connections to the D15 instances, both ABAP and Java, were broken. However, the components running on the se01 server were not affected.

The LifeKeeper log during the failure is shown in Example 6-5.

Example 6-5 Cluster management log from the failure to the recovery

```
COMMUNICATION TO se02 BY 192.168.234.1/192.168.234.2 FAILED AT: Thu Mar
13
    18:00:27 CET 2008
COMMUNICATION TO se02 BY 9.153.165.96/9.153.165.97 FAILED AT: Thu Mar
13
    18:00:27 CET 2008
COMMUNICATIONS failover from system "se02" will be started. AT: Thu Mar
13
    18:00:27 CET 2008
COMMUNICATIONS se02 FAILED AT Thu Mar 13 18:00:27 CET 2008
Thu Mar 13 18:00:27 CET 2008 FAILOVER RECOVERY OF MACHINE se02 STARTED
Thu Mar 13 18:00:28 CET 2008 FAILOVER RECOVERY OF MACHINE se02 FINISHED
COMMUNICATION TO se02 BY 192.168.234.1/192.168.234.2 RESTORED AT: Thu
Mar 13
    18:03:09 CET 2008
COMMUNICATION TO se02 BY 9.153.165.96/9.153.165.97 RESTORED AT: Thu Mar
13
    18:03:09 CET 2008
COMMUNICATIONS se02 RESTORED AT Thu Mar 13 18:03:09 CET 2008
LifeKeeper: Beginning automatic switchback check for resources from
"se02" at:
    Thu Mar 13 18:03:10 CET 2008
LifeKeeper: Finished automatic switchback check for resources from
"se02" at:
    Thu Mar 13 18:04:01 CET 2008
RESOURCE PROTECTION ACTIVATED FOR se02 AT:
    Thu Mar 13 18:04:02 CET 2008
LifeKeeper: COMM_UP to machine se02 done.
Thu Mar 13 18:05:34 CET 2008 quickCheck: ERROR 109656: Calling
sendevent for resource "SAP-RDB_10" on server "se01."
RECOVERY class=sap event=repenq_recover name=RDB-10 STARTING AT: Thu
Mar 13
    18:05:35 CET 2008
/opt/LifeKeeper/bin/recover: resource "SAP-RDB_10" with id "RDB-10" has
    experienced failure event "sap,repenq_recover"
/opt/LifeKeeper/bin/recover: attempting recovery using resource
"SAP-RDB_10"
    after failure by event "sap,repenq_recover" on resource
"SAP-RDB_10"
```

```

Thu Mar 13 18:05:35 CET 2008 repenq_recover: BEGIN recovery of
"SAP-RDB_10" on server "se01."
Thu Mar 13 18:05:39 CET 2008 repenq_recover: END successful recovery of
resource "SAP-RDB_10" on server "se01."
/opt/LifeKeeper/bin/recover: recovery succeeded after event
      "sap,repnq_recover" using recovery at resource "SAP-RDB_10"
on
      failing resource "SAP-RDB_10"
RECOVERY class=sap event=repnq_recover name=RDB-10 ENDING AT: Thu Mar
13
      18:05:39 CET 2008

```

During the failure, there were no recorded errors in SAP log system. Only sessions in server se02 were lost.

6.2.3 Failure of the ABAP central services

In the event of failure of one of the two components (message server or enqueue server) of the central services instances ABAP or Java, LifeKeeper restarts the full central services instance, because the enqueue and the message server restart are not supported. More information about this can be found in SAP Note 768727 in the SAP Marketplace. See Table 6-4.

Table 6-4 Summary of ASCS10 instance test failure

Purpose	Simulate a failure of the ABAP ASCS10 instance.
Status before the test	Servers se01 and se02 were up and running: <ul style="list-style-type: none"> ▶ DVEBMS12 central instance was running in se01. ▶ ASCS10 ABAP instance was running in se01. ▶ SCS11 Java instance was running in se01. ▶ RDB database was running in se01. ▶ D15 application instance was running in se02. ▶ ERS28 and ERS29 enqueue replication instances were running in se02.
Execution steps	Kill a component of the ASCS10 instance.
Expected results	After the kill of a component of the ASCS10 instance, LifeKeeper would restart it.
Status check	Verification of the status of the ABAP message server and the enqueue server.
Behavior	LifeKeeper for Linux detected the failure of the component and restarted both central services instances.

In addition to the recovery of the ABAP central services instance, this test demonstrated the benefit of the enqueue replication server instance ERS28 running on server se02. Lock entries were generated with transactions SMLG and DB13 as shown in Figure 6-8.

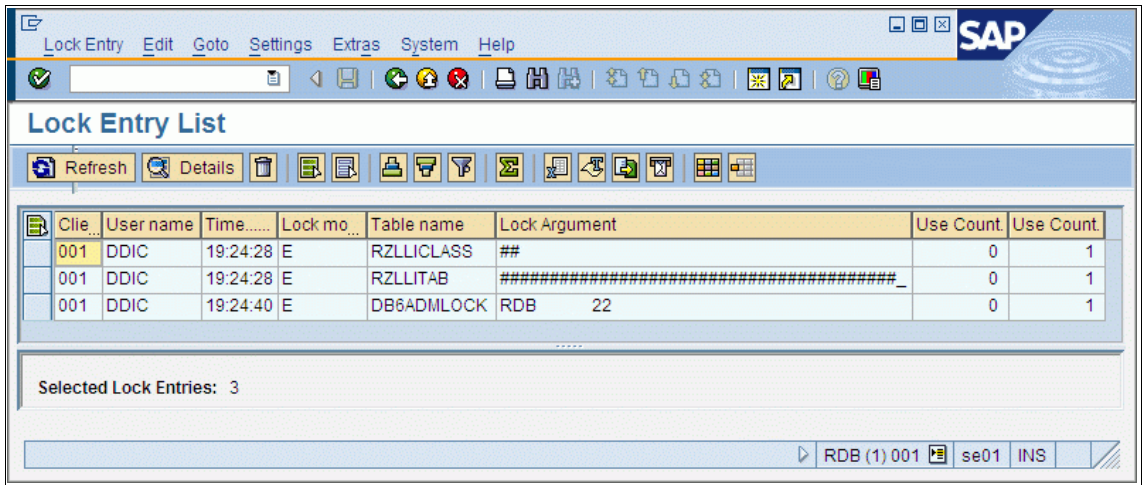


Figure 6-8 Lock entries in SAP RDB system

Before invoking failure of the ASCS10 component, the enqueue replication server status was verified as shown in Example 6-6.

Example 6-6 Replication verification

```
se02:rdbadm 52> ensmon -H sesap -I 10 2 | head
Try to connect to host sesap service sapdp10
get replinfo request executed successfully
Replication is enabled in server, repl. server is connected
Replication is active
=====
```

These processes composed the ASCS10 instance in the test environment:

```
ms.sapRDB_ASCS10 pf=/usr/sap/RDB/SYS/profile/RDB_ASCS10_sesap
en.sapRDB_ASCS10_sesap pf=/usr/sap/RDB/SYS/profile/RDB_ASCS10_sesap
```

The command **kill -9** was used to terminate the processes immediately; first the message server and after its recovery, the enqueue server. LifeKeeper detected the failure and restarted the central services instances automatically. The section of the LifeKeeper log regarding the message server failure is shown in Example 6-7.

Example 6-7 Message server failure and recovery

```
Fri Mar 14 09:21:33 CET 2008 quickCheck: ERROR 109654: Calling
sendevent for resource "SAP-RDB_10" on server "se01."
RECOVERY class=sap event=recover name=RDB-10 STARTING AT: Fri Mar 14
09:21:33
      CET 2008
/opt/LifeKeeper/bin/recover: resource "SAP-RDB_10" with id "RDB-10" has
      experienced failure event "sap,recover"
/opt/LifeKeeper/bin/recover: attempting recovery using resource
"SAP-RDB_10"
      after failure by event "sap,recover" on resource "SAP-RDB_10"
Fri Mar 14 09:21:34 CET 2008 recover: BEGIN recovery of "SAP-RDB_10" on
server "se01."
```

Stopping the SAP instance SCS11

```
-----
Shutdown-Log is written to /home/rdbadm/stopsap_SCS11.log
Instance on host se01 stopped
Waiting for cleanup of resources.....
_RetValVAL_=0
```

Stopping the SAP instance ASCS10

```
-----
Shutdown-Log is written to /home/rdbadm/stopsap_ASCS10.log
Instance on host se01 stopped
Waiting for cleanup of resources.....
_RetValVAL_=0
```

Starting SAP-Collector Daemon

```
-----
09:21:52 14.03.2008 LOG: Effective User Id is root
*****
* This is Saposcol Version COLL 20.94 700 - v2.00, AMD/Intel x86_64
with Linux, 2007/02/16
* Usage: saposcol -l: Start OS Collector
*          saposcol -k: Stop OS Collector
*          saposcol -d: OS Collector Dialog Mode
*          saposcol -s: OS Collector Status
* The OS Collector (PID 24535) is already running .....
```

```
*****
*
saposcol already running
Running /usr/sap/RDB/SYS/exe/run/startj2eedb
/usr/sap/RDB/SYS/exe/run/startj2eedb completed successfully
```

```

Starting SAP Instance SCS11
-----
Startup-Log is written to /home/rdbadm/startsap_SCS11.log
Instance Service on host se01 started
Instance on host se01 started

_RetValVAL_=0

Starting SAP-Collector Daemon
-----
09:22:25 14.03.2008 LOG: Effective User Id is root
*****
* This is Saposcol Version COLL 20.94 700 - v2.00, AMD/Intel x86_64
with Linux, 2007/02/16
* Usage: saposcol -l: Start OS Collector
*         saposcol -k: Stop OS Collector
*         saposcol -d: OS Collector Dialog Mode
*         saposcol -s: OS Collector Status
* The OS Collector (PID 24535) is already running .....
```

```

*****
*
* saposcol already running

Starting SAP Instance ASCS10
-----
Startup-Log is written to /home/rdbadm/startsap_ASCS10.log
Instance Service on host se01 started
Instance on host se01 started

_RetValVAL_=0
Fri Mar 14 09:22:58 CET 2008 recover: END successful recovery of
resource "SAP-RDB_10" on server "se01."
/opt/LifeKeeper/bin/recover: recovery succeeded after event
"sap,recover" using
        recovery at resource "SAP-RDB_10" on failing resource
"SAP-RDB_10"
RECOVERY class=sap event=recover name=RDB-10 ENDING AT: Fri Mar 14
09:22:58 CET
        2008

```

Note: Both central services instances were restarted because they belong to the same resource group, as recommended by SAP in the installation manual.

The copy of the enqueue table created by the enqueue server was used to recreate the lock entries so the running transactions were not reset.

6.2.4 Failure of the Java central services

The Java central services instance provides the message and enqueue server for the Java application server. The ABAP central services instance model was based on it and they perform similar tasks for the SAP system. See Table 6-5.

Table 6-5 Java central service test summary

Purpose	Simulate a failure of the Java central instance.
Status before the test	Servers se01 and se02 were up and running: <ul style="list-style-type: none">▶ DVEBMS12 central instance was running in se01.▶ ASCS10 ABAP instance was running in se01.▶ SCS11 Java instance was running in se01.▶ RDB database was running in se01.▶ D15 application instance was running in se02.▶ ERS28 and ERS29 enqueue replication instances were running in se02.
Execution steps	Terminate the Java central instance process.
Expected results	After the kill of a component of the SCS11 instance, LifeKeeper would restart it.
Status check	Verification of the status of Java message, enqueue, and replication servers.
Behavior	LifeKeeper for Linux detected the failure of the component and restarted both central services instances.

The Java enqueue server replication was verified with the command shown in Example 6-8.

Example 6-8 Enqueue server replication status

```
se02:rdbadm 63> ensmon -H se01 -I 11 2 | head
Try to connect to host se01 service sapdp11
get replinfo request executed successfully
```

```
Replication is enabled in server, repl. server is connected
Replication is active
```

```
=====
```

The message server was verified with the command shown in Example 6-9.

Example 6-9 Java message server availability

```
se02:rdbadm 64> telnet sesap 3311
Trying 9.153.165.98...
Connected to sesap.
Escape character is '^['.
```

Lock entries were created in the Java instance as shown in Figure 6-9.

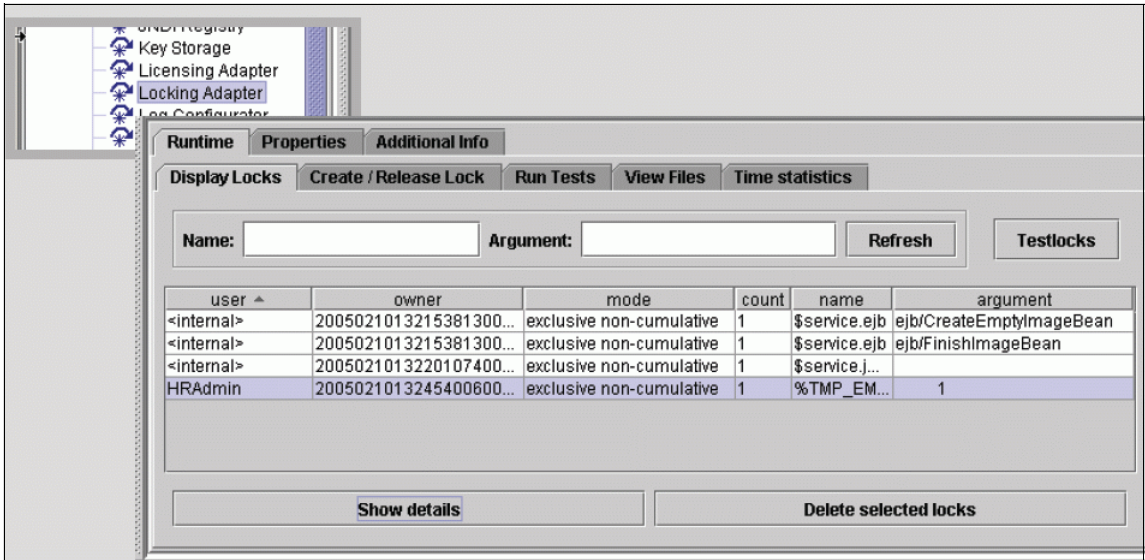


Figure 6-9 Java lock entries

The Java enqueue server was terminated with the **kill -9** command, and as with the ABAP instance, LifeKeeper detected the failure, restarting both central services instances.

The lock entries were recovered from the copy in the enqueue replication server using the replication enqueue server copy table.

6.2.5 Failure of the central and application instances

The central instance of the SAP NetWeaver 7.0 can be replicated with the installation of another application server (former dialog instances). The central instance of the test environment for this book was named DVEBMGS12 and the installed application server D15.

To ascertain the impact in the event of a central instance or application instance failure, the central instance was stopped. See Table 6-6.

Table 6-6 Central instance failure simulation

Purpose	Simulate a failure of the central instance application.
Status before the test	Servers se01 and se02 were up and running: <ul style="list-style-type: none">▶ DVEBMGS12 central instance was running in se01.▶ ASCS10 ABAP instance was running in se01.▶ SCS11 Java instance was running in se01.▶ RDB database was running in se01.▶ D15 application instance was running in se02.▶ ERS28 and ERS29 enqueue replication instances were running in se02.
Execution steps	Stop central instance.
Expected results	Process running in DVEBMSG12 instances is cancelled. The SAP system is still available through the D15 Java and ABAP instances.
Status check	Verification of the status DVEBMGS12 instance.
Behavior	LifeKeeper for Linux detected the failure application instance and restarted it. All processes running in the stopped instance were cancelled.

The status of the central instance (se01_RDB_12) and the application instance (se02_RDB_15) before the test are shown in Figure 6-10.

Server Name	Name	Message Types	Server Status
se01_RDB_12	se01	Dialog Batch Update Upd2 Spool ICM J2EE	Active
se02_RDB_15	se02	Dialog Batch Update Upd2 Spool ICM J2EE	Active

Figure 6-10 Active Instances

Java instances were also available as shown in Figure 6-11.

Instance se02_RDB_15		All processes running	
Host:	se02	OS:	Linux (amd64) 2.6.16.46-0.12-smp
dispatcher		Running	
VM	system properties...	Cluster	
PID:	32333	Node ID:	155175200
Name:	IBM J9SE VM	Kernel	7.00
Vendor:	IBM Corporation	Version:	PatchLevel
Version:	2.2	HTTP Port:	51500
VM Parameters		HTTPS Port:	51501
		P4 Port:	51504
		Telnet Port:	51508

Instance se01_RDB_12		All processes running	
Host:	se01	OS:	Linux (amd64) 2.6.16.46-0.12-smp
dispatcher		Running	
VM	system properties...	Cluster	
PID:	7327	Node ID:	127218100
Name:	IBM J9SE VM	Kernel	7.00
Vendor:	IBM Corporation	Version:	PatchLevel
Version:	2.2	HTTP Port:	51200
VM Parameters		HTTPS Port:	51201
		P4 Port:	51204
		Telnet Port:	51208

Figure 6-11 Java Instances status

In order to review the outcome of the failure, LifeKeeper was prevented from automatically starting the stopped instance immediately. The SAP resources in LifeKeeper were put on hold in the administration console. For further information regarding LifeKeeper for Linux administration, refer to Chapter 7., “Administering the cluster” on page 223.

Central instance DVEBMGS12 was stopped with the default **stopsap** command. The SAP RDB system was still available through the logon group Redbook, and the transaction SM51 server status is shown in Figure 6-12.

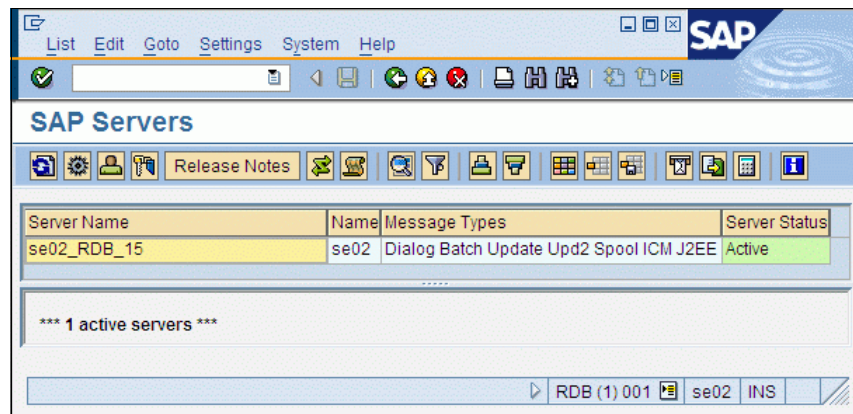


Figure 6-12 Remaining application after central instance shutdown

All functions of the SAP system were still available without the central instance, except for the Software Deployment Manager, but this is not considered a critical component for production environments.

Sessions connected to the instance DVEBMGS12 were broken, both in ABAP and Java instances.

6.2.6 Failure of the database system

The database system used in the test environment was DB2 for Linux, UNIX, and Windows, on a UNIX database.

DB2 has a command that can be used to kill all processes for an instance. This **db2_kill** command terminates all database process and report the condition to the administration log.

Rather than start the database in the same server, this failure scenario also blocked DB2 admin user, db2rdm, forcing LifeKeeper to move the resources to the standby server. See Table 6-7.

Table 6-7 Summary of the database failure simulation

Purpose	Simulate a failure of the database system.
Status before the test	Servers se01 and se02 were up and running: <ul style="list-style-type: none">▶ DVEBMS12 central instance was running in se01▶ ASCS10 ABAP instance was running in se01▶ SCS11 Java instance was running in se01▶ RDB database was running in se01▶ D15 application instance was running in se02▶ ERS28 and ERS29 enqueue replication instances were running in se02
Execution steps	Terminate the DB2 processes with the db2_kill command and lock the db2rdm user.
Expected results	LifeKeeper for Linux would have detected the failure and restarted the database.
Status check	Verification of the database status.
Behavior	LifeKeeper for Linux detected the failure of the standby server and restarted the instances in se02 when the server was ready.

The DB2 status was checked before the failure as shown in Example 6-10.

Example 6-10 DB2 threads running on server se01

```
se01:db2rdb 52> db2 list applications
```

```
Auth Id Application  Appl.    Application Id
DB      # of
        Name        Handle
Name    Agents
-----
-----
-----
SAPRDB  dw.sapRDB_D15  92      9.153.165.97.60894.080314104741
RDB      1
SAPRDB  dw.sapRDB_DVEB 151     *LOCAL.db2rdb.080314112645
RDB      1
SAPRDB  dw.sapRDB_D15  98      9.153.165.97.62430.080314104741
RDB      1
SAPRDB  dw.sapRDB_DVEB 157     *LOCAL.db2rdb.080314112651
RDB      1
SAPRDBDB db2jcc_applica 170     9.153.165.98.22243.080314112718
RDB      1
```

SAPRDB	dw.sapRDB_D15	91	9.153.165.97.60382.080314104740
RDB	1		
SAPRDB	dw.sapRDB_D15	97	9.153.165.97.62174.080314104741
RDB	1		
SAPRDB	dw.sapRDB_DVEB	156	*LOCAL.db2rdb.080314112649
RDB	1		
SAPRDBDB	db2jcc_applica	169	9.153.165.98.21475.080314112712
RDB	1		
SAPRDB	dw.sapRDB_DVEB	162	*LOCAL.db2rdb.080314112656
RDB	1		
SAPRDB	dw.sapRDB_D15	96	9.153.165.97.61918.080314104740
RDB	1		
SAPRDBDB	db2jcc_applica	109	9.153.165.97.10954.080314104810
RDB	1		
SAPRDB	dw.sapRDB_DVEB	155	*LOCAL.db2rdb.080314112648
RDB	1		
SAPRDB	dw.sapRDB_D15	102	9.153.165.97.63710.080314104742
RDB	1		
SAPRDB	dw.sapRDB_DVEB	161	*LOCAL.db2rdb.080314112655
RDB	1		
SAPRDB	dw.sapRDB_D15	95	9.153.165.97.61662.080314104740
RDB	1		
SAPRDB	dw.sapRDB_DVEB	154	*LOCAL.db2rdb.080314112650
RDB	1		
SAPRDB	dw.sapRDB_D15	101	9.153.165.97.63454.080314104742
RDB	1		
SAPRDB	dw.sapRDB_DVEB	160	*LOCAL.db2rdb.080314112654
RDB	1		
SAPRDB	dw.sapRDB_D15	94	9.153.165.97.61406.080314104740
RDB	1		
SAPRDBDB	db2jcc_applica	140	9.153.165.97.28058.080314111246
RDB	1		
SAPRDB	dw.sapRDB_DVEB	153	*LOCAL.db2rdb.080314112647
RDB	1		
DB2RDB	javaw.exe	278	G999A15A.KE07.00E104121732
RDB	1		
SAPRDB	dw.sapRDB_D15	100	9.153.165.97.62942.080314104742
RDB	1		
SAPRDB	dw.sapRDB_DVEB	159	*LOCAL.db2rdb.080314112652
RDB	1		
SAPRDBDB	db2jcc_applica	172	9.153.165.98.22755.080314112736
RDB	1		
SAPRDB	dw.sapRDB_D15	93	9.153.165.97.61150.080314104740
RDB	1		

```

SAPRDB dw.sapRDB_DVEB 152    *LOCAL.db2rdb.080314112646
RDB      1
SAPRDB dw.sapRDB_D15 99      9.153.165.97.62686.080314104742
RDB      1
SAPRDBDB db2jcc_applica 112   9.153.165.97.12490.080314104824
RDB      1
SAPRDB dw.sapRDB_DVEB 158    *LOCAL.db2rdb.080314112653
RDB      1

```

When the DB2 RDB database was terminated, the SAP workprocess entered into reconnecting state. Figure 6-13 shows this section of the SAP log systems.

Time	Type	Nr	Clt	User	TCode	Priority	Grp	N	Text
13:08:14	S-A	000		rdbadm			E0	7	Error 000104 : Connection reset by peer in Module rslgsend(027)
13:27:44	S-A	000		rdbadm			E0	7	Error 000104 : Connection reset by peer in Module rslgsend(027)
13:28:44	S-A	000		rdbadm			E0	7	Error 000104 : Connection reset by peer in Module rslgsend(027)
13:30:44	S-A	000		rdbadm			E0	7	Error 000104 : Connection reset by peer in Module rslgsend(027)
13:34:44	S-A	000		rdbadm			E0	7	Error 000104 : Connection reset by peer in Module rslgsend(027)
13:42:44	S-A	000		rdbadm			E0	7	Error 000104 : Connection reset by peer in Module rslgsend(027)
14:15:08	DIA	000	000	SAPSYS			BY	M	SQL error -1224. Work processes in reconnect status
14:15:18	SPO	010					BV	4	Work process is in reconnect status
14:15:18	DIA	001					BV	4	Work process is in reconnect status
14:15:18	DIA	000					BV	4	Work process is in reconnect status
14:15:28	DIA	002					BV	4	Work process is in reconnect status
14:15:29	DIA	003					BV	4	Work process is in reconnect status
14:15:38	DIA	002					BZ	Y	Unexpected return value 29 when calling up DbS1
14:15:38	DIA	002					F6	H	Database error: TemSe->RTAB-S/G(8)->4 for table TCPSBUILD key
14:15:38	DIA	002					CP	R	No Code Page Conversion '4103'->'1100' : CCC->CCC
14:15:38	DIA	002					BZ	Y	Unexpected return value 29 when calling up DbS1
14:15:38	DIA	002					F6	H	Database error: TemSe->RTAB-S/G(8)->4 for table TCPSBUILD key 4103T*
14:15:38	DIA	002					CP	R	No Code Page Conversion '4103'->'1100' : CCC->CCC
14:15:39	DIA	003					BZ	Y	Unexpected return value 29 when calling up DbS1
14:15:39	DIA	003					F6	H	Database error: TemSe->RTAB-S/G(8)->4 for table TCPSBUILD key 4103T*
14:15:39	DIA	003					CP	R	No Code Page Conversion '4103'->'1100' : CCC->CCC
14:15:39	DIA	003					BZ	Y	Unexpected return value 29 when calling up DbS1
14:15:39	DIA	003					F6	H	Database error: TemSe->RTAB-S/G(8)->4 for table TCPSBUILD key 4103T*
14:15:39	DIA	003					CP	R	No Code Page Conversion '4103'->'1100' : CCC->CCC
14:16:26	DIA	003					BY	Y	Work process has left reconnect status
14:16:26	SPO	010					BY	Y	Work process has left reconnect status
14:16:26	DIA	002					BY	Y	Work process has left reconnect status
14:16:27	DIA	001					BY	Y	Work process has left reconnect status
14:19:02	DIA	000					BY	Y	Work process has left reconnect status

Figure 6-13 SAP RDB system log during database failure

LifeKeeper quickly detected the failure and tried to restart database in server se01. Because the db2rdm user was revoked it failed and LifeKeeper moved the resources to se02 server and started the database there.

Since the database hostname (sedb) was moved as part of the resource group, SAP system just reconnected to the DB2 RDB database.

All SAP sessions were terminated during the unavailability of the database.

6.2.7 Failure of the NFS Server

The NFS server is an important software component in the SAP system because the applications servers mount core filesystem using this protocol.

This test aims to verify the behavior of the LifeKeeper for Linux and SAP NetWeaver applications servers in the event of a NFS failure.

Table 6-8 NFS Server failure in the active server

Purpose	Simulate a failure of the NFS server.
Status before the test	Servers se01 and se02 were up and running: <ul style="list-style-type: none">▶ DVEBMS12 central instance was running in se01.▶ ASCS10 ABAP instance was running in se01.▶ SCS11 Java instance was running in se01.▶ RDB database was running in se01.▶ D15 application instance was running in se02.▶ ERS28 and ERS29 enqueue replication instances were running in se02.
Execution steps	Terminate NFS Server daemon.
Expected results	LifeKeeper for Linux would have detected the failure and restarted the daemon. The mount points in the application server running on se02 should be re-established.
Status check	Verification of NFS exported file systems in the application server.
Behavior	LifeKeeper for Linux detected the failure of the NFS Server and restarted it. The NFS mounting points were re-established. The failure did not affect the SAP RDB system.

The NFS server of the test environment was running in the same server as the central instance, se01. The NFS daemons were terminated with the **kill -9** command as shown in Example 6-11.

Example 6-11 Terminating NFS Server

```
se01:~ # ps -ef | grep -i nfs
root      5568      9  0 10:45 ?        00:00:00 [nfsd4]
root      5569      1  0 10:45 ?        00:00:00 [nfsd]
root      5570      1  0 10:45 ?        00:00:00 [nfsd]
root      5571      1  0 10:45 ?        00:00:01 [nfsd]
root      5572      1  0 10:45 ?        00:00:00 [nfsd]
root      25916  7791  0 17:16 pts/3    00:00:00 grep -i nfs
se01:~ # kill -9 5568 5569 5570 5571 5572
```

LifeKeeper detected the failure and restarted the NFS server successfully. During the failure, the unavailability of the NFS mount points does not affect the SAP application server running on server se02.

6.2.8 Planned outages

DB2 9 for Linux, UNIX, and Windows allows for the co-existence of multiple DB2 versions and fix packs. Although previous versions of DB2 provided this ability with alternative fix pack images, this was not a supported installation for SAP. With DB2 9, customers can install multiple versions of DB2 on the same physical machine. Each copy can be updated without affecting the other. This provides flexibility in testing and deploying new fix packs.

SAP also provides a similar feature. During the start of the instances, SAP access the /sapmnt/<SID>/exe directory and copies important files to the local instance directory using the **sapcpe** command.

For operating system and hardware updates, it is possible to switch all applications to one server, update it, and do the same for the other server(s). Table 6-9 below shows three planned outage scenarios:

Table 6-9 *Planned outage scenarios*

Scenario	Description
Normal startup of the system	The startup process for all components together
Normal shutdown of the system	The shutdown process for all components together
Applications switch	The move of the resource groups from one server to another

The first two scenarios are the usual start and stop of the system, and further information about how to perform these actions is in Chapter 7., “Administering the cluster” on page 223. The third one is the scenario used to update the system without a full unavailability of the application.



Administering the cluster

In this chapter, we discuss administrative tasks and operation aspects of a high availability scenario based on SteelEye LifeKeeper for Linux, SAP NetWeaver, and SUSE Linux Enterprise Server.

The content is organized as grouped components that relate to these topics and explains how they integrate with each other:

- ▶ “Base operating system”
- ▶ “SteelEye LifeKeeper administration”
- ▶ “Customizing of LifeKeeper parameters”
- ▶ “Maintenance during uptime”
- ▶ “Backup and restore”

7.1 Base operating system

The section discusses tasks that relate to the base operating system:

- ▶ “Redundant Local Area Network connection”
- ▶ “Redundant Storage Area Network connection”
- ▶ “Mirroring data across storage sub-systems”
- ▶ “Enhancing a file system on the shared storage”
- ▶ “Frequent file system checks”

7.1.1 Redundant Local Area Network connection

When using the bonding module to configure redundant network interfaces, an entry in the proc file system is created for every bonded interface, such as:

- ▶ `/proc/net/bonding/bond0`
- ▶ `/proc/net/bonding/bond1`

Example 7-1 shows how these entries can be listed by using the **cat** command, for example. They contain detailed information about the status of the virtual interface and the physical interfaces.

Example 7-1 Content of `/proc/net/bonding/bond0`

Ethernet Channel Bonding Driver: v2.6.5 (November 4, 2005)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: **eth0**
MII Status: up
MII Polling Interval (ms): 0
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: **eth0**
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:0e:61:1ab:e6:77

Slave Interface: **eth2**
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:07:e9:0d:b9:ec

For monitoring, the value of the Currently Active Slave must be tracked for any changes. Alternatively, it is possible to set a Primary Slave. If the Primary Slave is not the Currently Active Slave, the bonding interface is in a failover situation and must be actioned to resolve it.

For testing, the physical interfaces can be disabled or enabled using the **up** and **down** options of the **ifconfig** command.

7.1.2 Redundant Storage Area Network connection

The Device-Mapper Multipath I/O sub-system verifies if a Storage Area Network communication path is working. The **multipathd** needs to be running. It must be enabled to start automatically at system startup (you can verify with **chkconfig multipathd** commands).

The status of all paths can be displayed with the **multipath -l** command. Example 7-2 shows detailed output about each path and its status.

All paths are [active][undef] — that means that they are available and working. If one or many paths fail, they show a status of [failed][faulty].

Example 7-2 Status output of the multipath command

```
sedb-1 (3600507680185000d3800000000000054) dm-12 IBM,2145
[size=128G][features=0][hwhandler=0]
\_ round-robin 0 [prio=0][active]
\_ 1:0:0:12 sdd 8:48 [active][undef]
\_ 2:0:0:12 sdl 8:176 [active][undef]
\_ round-robin 0 [prio=0][enabled]
\_ 1:0:1:12 sdh 8:112 [active][undef]
\_ 2:0:1:12 sdp 8:240 [active][undef]
sedb-0 (3600507680185000d3800000000000053) dm-11 IBM,2145
[size=128G][features=0][hwhandler=0]
\_ round-robin 0 [prio=0][active]
\_ 1:0:1:11 sdg 8:96 [active][undef]
\_ 2:0:1:11 sdo 8:224 [active][undef]
\_ round-robin 0 [prio=0][enabled]
\_ 1:0:0:11 sdc 8:32 [active][undef]
\_ 2:0:0:11 sdk 8:160 [active][undef]
sesap-1 (3600507680185000d3800000000000056) dm-16 IBM,2145
[size=64G][features=0][hwhandler=0]
\_ round-robin 0 [prio=0][active]
\_ 1:0:0:14 sdf 8:80 [active][undef]
\_ 2:0:0:14 sdn 8:208 [active][undef]
\_ round-robin 0 [prio=0][enabled]
```

```

\_ 1:0:1:14 sdj 8:144 [active][undef]
\_ 2:0:1:14 sdr 65:16 [active][undef]
SIBM-ESXSST973401LC_F3LB0EEGN00007631J2LXdm-10 IBM-ESXS,ST973401LC
[size=68G][features=0][hwhandler=0]
\_ round-robin 0 [prio=0][active]
\_ 0:0:1:0 sdb 8:16 [active][undef]
sesap-0 (3600507680185000d3800000000000055) dm-13 IBM,2145
[size=64G][features=0][hwhandler=0]
\_ round-robin 0 [prio=0][active]
\_ 1:0:1:13 sdi 8:128 [active][undef]
\_ 2:0:1:13 sdq 65:0 [active][undef]
\_ round-robin 0 [prio=0][enabled]
\_ 1:0:0:13 sde 8:64 [active][undef]
\_ 2:0:0:13 sdm 8:192 [active][undef]

```

7.1.3 Mirroring data across storage sub-systems

The Software RAID based mirrors can be checked frequently through the mdadm process. The configuration file `/etc/mdadm.conf` holds configuration information for individual arrays and some options for monitoring and notification:

- ▶ **MAILADDR:** This is an Internet style mail address that needs to be reachable from the system. This mail address receives alerts from the Software RAID, for example, when a disk fails,
- ▶ **MAILFROM:** This is another Internet style mail address which is used for sending out notifications and
- ▶ **PROGAM:** This is a program that should be run when an event happens.

Example 7-3 shows sample configuration file used in the test scenario.

Example 7-3 Content of configuration file `/etc/mdadm.conf`

```

ARRAY /dev/md0
    level=raid1
    num-devices=2
    UUID=eecd5458:a66a1f98:4eeda444:9de774a3

ARRAY /dev/md1
    level=raid1
    num-devices=2
    UUID=e996777e:484327ff:cb25cebc:77e7d0df

MAILFROM raid@se01.redbook.lan
MAILADDR operating@redbook.lan

```

The status of a Software RAID array can be seen in the virtual file `/proc/mdstat`. It shows information about all active RAID arrays. In a cluster, these arrays can move. To see the status of all RAID arrays in a cluster, the status has to be verified on all nodes individually. A RAID array must never be active on more than one node.

Example 7-4 shows the content of one node in the test scenario at a time when both RAID arrays have been active at the same time.

Example 7-4 Content of /proc

```
Personalities : [raid1]
md1 : active raid1 dm-18[0] dm-17[1]
      67103360 blocks [2/2] [UU]

md0 : active raid1 dm-14[0] dm-15[2]
      134214912 blocks [2/2] [UU]

unused devices: <none>
```

Another way to view detailed information about an array is by issuing the **mdadm** command. Example 7-5 is the output from command **mdadm --detail /dev/md1**.

Note: When Software RAID arrays become integrated into a LifeKeeper resource hierarchy, the `mdadm` daemon is disabled. LifeKeeper performs monitoring and notification for all cluster resources, including the Software RAID arrays. When all Software RAID arrays are under control of LifeKeeper, `mdadm` can be disabled. Internal disks can be mirrored with Software RAID but cannot be under cluster control. To monitor these, or for additional monitoring, the `mdadm` has to be integrated with the LifeKeeper hierarchy. “Integrating `mdadm` in the LifeKeeper hierarchy” on page 243 gives an example of `mdadm` integration into a resource hierarchy.

Example 7-5 Example of mdadm detailed output

```
/dev/md1:
  Version : 00.90.03
  Creation Time : Thu Feb 21 15:53:46 2008
    Raid Level : raid1
    Array Size : 67103360 (63.99 GiB 68.71 GB)
  Used Dev Size : 67103360 (63.99 GiB 68.71 GB)
    Raid Devices : 2
    Total Devices : 2
  Preferred Minor : 1
    Persistence : Superblock is persistent

    Update Time : Mon Mar 10 20:20:15 2008
      State : clean
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0


    UUID : eecd5458:a66a1f98:4eeda444:9de774a3
    Events : 0.656549
```

Number	Major	Minor	RaidDevice	State	
0	253	18	0	active sync	/dev/dm-18
1	253	17	1	active sync	/dev/dm-17

In the examples above, the RAID array md1 consists of two disks, dm-18 and dm-17; and md0 consists of dm-14 and dm-15. The output shows the major and minor device numbers. Alternatively, they can be found by looking for the device id of /dev/dm-18 using the `ls` command. Example 7-6 shows the output of the command in the test scenario.

Example 7-6 Get device id from device path

```
se01:~ # ls -l /dev/dm-18
brw-r----- 1 root disk 253, 18 Mar  7 16:12 /dev/dm-18
se01:~ #
```

This shows a major number of 253 and minor number of 18. This is also given in the detailed mdadm output. The device mapper module creates entries for all devices in the directory /dev/mapper. Now the device name can be located by searching for the device identified by major number 253 and minor number 18 in /dev/mapper.

Example 7-7 shows the output from the /dev/mapper directory. From this example, the major number 253 and minor number 18 point to the name sesap-0-part1.

Example 7-7 Device entries in the /dev/mapper directory

```
se01:~ # ls -l /dev/mapper
total 0
brw----- 1 root root 253, 10 Mar  7 16:12 SIBM-ESXSST973401LC_F...
lrwxrwxrwx 1 root root      16 Mar  7 16:12 control -> ../device-mapper
brw----- 1 root root 253, 11 Mar  7 16:12 sedb0
brw----- 1 root root 253, 14 Mar  7 16:12 sedb0-part1
brw----- 1 root root 253, 12 Mar  7 16:12 sedb1
brw----- 1 root root 253, 15 Mar  7 16:12 sedb1-part1
brw----- 1 root root 253, 32 Mar 10 17:58 sedbvg-RDB_db2dump1v
brw----- 1 root root 253, 33 Mar 10 17:58 sedbvg-RDB_db2rdb1v
brw----- 1 root root 253, 30 Mar 10 17:58 sedbvg-RDB_log_dir1v
brw----- 1 root root 253, 19 Mar 10 17:57 sedbvg-RDB_sapdata1lv
brw----- 1 root root 253, 20 Mar 10 17:58 sedbvg-RDB_sapdata2lv
brw----- 1 root root 253, 31 Mar 10 17:58 sedbvg-RDB_sapdata3lv
brw----- 1 root root 253, 27 Mar 10 17:58 sedbvg-RDB_sapdata4lv
brw----- 1 root root 253, 26 Mar 10 17:58 sedbvg-RDB_saptemp1lv
brw----- 1 root root 253, 34 Mar 10 17:58 sedbvg-db2_2lv
brw----- 1 root root 253, 29 Mar 10 17:58 sedbvg-db2binlv
brw----- 1 root root 253, 21 Mar 10 17:58 sedbvg-db2rdblv
brw----- 1 root root 253, 13 Mar  7 16:12 sesap-0
brw----- 1 root root 253, 18 Mar  7 16:12 sesap-0-part1
brw----- 1 root root 253, 16 Mar  7 16:12 sesap-1
brw----- 1 root root 253, 17 Mar  7 16:12 sesap-1-part1
brw----- 1 root root 253, 25 Mar  7 17:49 sesapvg-ascsl0lv
brw----- 1 root root 253, 24 Mar  7 17:50 sesapvg-dvebmsg12lv
brw----- 1 root root 253, 22 Mar  7 17:49 sesapvg-exportsapmntlv
brw----- 1 root root 253, 23 Mar  7 17:49 sesapvg-scs11lv
brw----- 1 root root 253, 28 Mar  7 17:50 sesapvg-ursaptranslv
brw----- 1 root root 253,  9 Mar  7 16:12 systemvg-exportsapmnt-
brw----- 1 root root 253,  0 Mar  7 16:12 systemvg-homelv
brw----- 1 root root 253,  1 Mar  7 16:12 systemvg-optlv
brw----- 1 root root 253,  2 Mar  7 16:12 systemvg-rootlv
brw----- 1 root root 253,  3 Mar  7 16:12 systemvg-swaplv
brw----- 1 root root 253,  4 Mar  7 16:12 systemvg-tmplv
brw----- 1 root root 253,  5 Mar  7 16:12 systemvg-usrlv
brw----- 1 root root 253,  7 Mar  7 16:12 systemvg-ursaplv
brw----- 1 root root 253,  8 Mar  7 16:12 systemvg-ursaprdblv
brw----- 1 root root 253,  6 Mar  7 16:12 systemvg-varlv
se01:~ #
```

Alternatively, the same entry can be found by looking into the directory /dev/disk/by-name, which contains alias names from the Device-Mapper Multipaths I/O module as shown in Example 7-8.

Example 7-8 Alias names from the Device-Mapper Multipaths I/O module

```
se01:~ # ls -l /dev/disk/by-name/se*
lrwxrwxrwx 1 root root 11 Mar  7 16:12 sedb0 -> ../../dm-11
lrwxrwxrwx 1 root root 11 Mar  7 16:12 sedb0-part1 -> ../../dm-14
lrwxrwxrwx 1 root root 11 Mar  7 16:12 sedb1 -> ../../dm-12
lrwxrwxrwx 1 root root 11 Mar  7 16:12 sedb1-part1 -> ../../dm-15
lrwxrwxrwx 1 root root 11 Mar  7 16:12 sesap-0 -> ../../dm-13
lrwxrwxrwx 1 root root 11 Mar  7 16:12 sesap-0-part1 -> ../../dm-18
lrwxrwxrwx 1 root root 11 Mar  7 16:12 sesap-1 -> ../../dm-16
lrwxrwxrwx 1 root root 11 Mar  7 16:12 sesap-1-part1 -> ../../dm-17
se01:~ #
```

The alias names can be used in the **mdadm** command. It is also possible to use the short names in the **mdadm** command, for example, /dev/dm-15 instead of /dev/disk/by-name/sedb1-part1.

When a failure occurs, the broken part of Software RAID goes into a state failed. This can be simulated manually by setting a drive to a faulty state through the **mdadm** command. This may be useful prior to applying maintenance to a whole storage subsystem. Example 7-9 shows setting a disk to a faulty state.

Example 7-9 Setting one disk of a mirror to faulty state

```
se01:~ # mdadm /dev/md0 -f /dev/disk/by-name/sedb1-part1
mdadm: set /dev/disk/by-name/sedb1-part1 faulty in /dev/md0

se01:~ #
```

The status goes to a faulty state immediately and its status can be seen in /proc/mdstat. Example 7-10 shows an output of a failed device:

Example 7-10 Sample output of /proc/mdstat with a failed device

```
se01:~ # cat /proc/mdstat
Personalities : [raid1]
md1 : active raid1 dm-18[0] dm-17[1]
      67103360 blocks [2/2] [UU]

md0 : active raid1 dm-15[2] (F) dm-14[0]
      134214912 blocks [2/1] [U_]
```



```
unused devices: <none>
se01:~ #
```

When a drive has gone to a faulty state, it can be reactivated by removing it from the array and re-adding it. This forces a full synchronization of that disk and can take a while, depending on the throughput and the amount of data.

In a Storage Area Network scenario, this is necessary if all paths to a storage subsystem have failed. Example 7-11 shows the commands for removing one mirror from a Software RAID device and re-adding it.

Example 7-11 Removing and re-adding a failed drive

```
se01:~ # mdadm /dev/md0 -r /dev/disk/by-name/sedb1-part1
mdadm: hot removed /dev/disk/by-name/sedb1-part1
se01:~ # mdadm /dev/md0 -a /dev/disk/by-name/sedb1-part1
mdadm: re-added /dev/disk/by-name/sedb1-part1
se01:~ #
```

After adding a drive to the array, the Software RAID instantly starts to synchronize the full mirror to the new disk. The `/proc/mdstat` entry displays the progress of the synchronization as shown in Example 7-12.

Example 7-12 Content of `/procM`

```
se01:~ # cat /proc/mdstat
Personalities : [raid1]
md1 : active raid1 dm-18[0] dm-17[1]
      67103360 blocks [2/2] [UU]

md0 : active raid1 dm-15[2] dm-14[0]
      134214912 blocks [2/1] [U_]
      [>.....] recovery = 1.8% (2440832/134214912)
      finish=40.0min speed=54822K/sec

unused devices: <none>
```

Tip: With the command `watch cat /proc/mdstat`, it is easy to watch the progress of a synchronization. The `watch` command refreshes the screen frequently; the default is every two seconds.

SteelEye LifeKeeper for Linux is constantly monitoring the status of a Software RAID device. The status is visible in the properties of the Software RAID resource in the LifeKeeper GUI.

Figure 7-1 shows an example of a synchronization in progress.

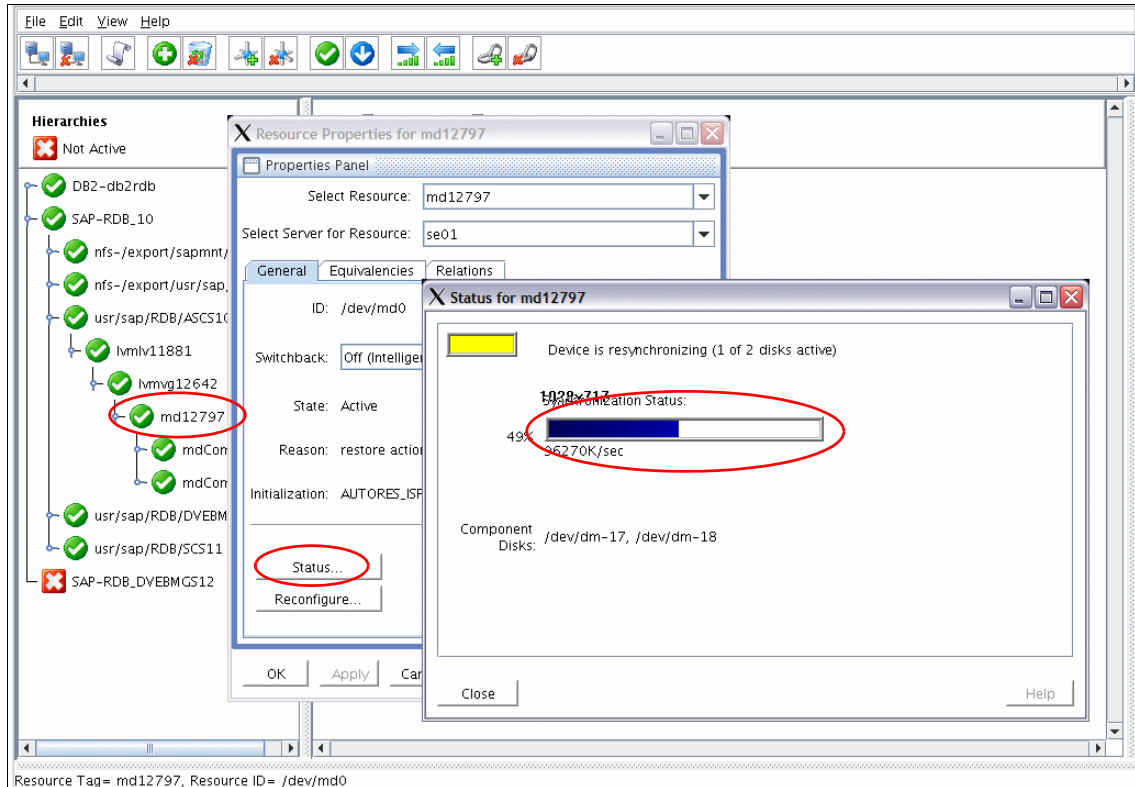


Figure 7-1 Software RAID synchronization status in the LifeKeeper GUI

The Software RAID module can be configured to use a particular bandwidth for synchronization. There are two values that can be changed:

- ▶ The *speed_limit_min* is the minimum bandwidth that the Software RAID module uses for synchronization. This affects other applications performing I/O operations on the underlying device and can result in performance degradation for an application that is running on it.
- ▶ The *speed_limit_max* is the maximum bandwidth a synchronization process may use. Depending on the Storage Area Network topology, one server can easily use up a lot of the storage performance.

Both parameters hold the throughput value in Kilobytes per second and they can be changed “on the fly” during an operation. They are represented as files in the /proc file systems containing the numerical values. The changes in the synchronization speed are visible soon in the /proc/mdstat entry.

Example 7-13 shows the default values of these settings, setting a value of 1 Gigabyte per second for both minimum and maximum to force a synchronization to use as much bandwidth as possible.

Example 7-13 Showing and setting the Software RAID speed limits.

```
se01:~ # ls /proc/sys/dev/raid
speed_limit_max speed_limit_min
se01:~ # cat /proc/sys/dev/raid/*
200000
1000
se01:~ # echo 1000000 > /proc/sys/dev/raid/speed_limit_max
se01:~ # echo 1000000 > /proc/sys/dev/raid/speed_limit_min
se01:~ # cat /proc/sys/dev/raid/*
1000000
1000000
```

Tip: Persistent change of these settings can be put into the `/etc/sysctl.conf` file. The variable names are `dev.raid.speed_limit_max` and `dev.raid.speed_limit_min`. The command `sysctl -a | grep raid` shows the content of the values. The `sysctl` command can be used during runtime instead of modifying the entries in the /proc file system, too.

The SteelEye LifeKeeper for Linux software monitors the availability of the drives belonging to a Software RAID array managed by the cluster. It shows failures in the LifeKeeper GUI and logs.

7.1.4 Enhancing a file system on the shared storage

Some file systems can be enhanced either online or offline. With the underlying Logical Volume Manager, the logical volume has to be enhanced before resizing the file system itself. The file system can be enhanced after the logical volume has been changed. Some file systems, for example ext2/ext3, can only be resized when they are offline.

To unmount a LifeKeeper managed file system, the resource hierarchy has to be set to the offline state down to the file system. The underlying volume group and Software RAID devices have to remain active.

When a file system is being resized online and while it is in use, no change within SteelEye LifeKeeper is necessary. Shared file systems can be resized without changing a LifeKeeper hierarchy.

If there is not enough space in a volume group, the volume group can be enhanced with another physical volume. This requires the following steps:

1. Creating new volumes on the affected storage sub-systems
2. Rescan of the SCSI bus on all cluster nodes
3. Creating a proper Device-Mapper Multipath I/O configuration for that drive and reloading the configuration
4. Partitioning the Multipath devices
5. Creating a Software RAID array
6. Adding the Software RAID array to the active volume group
7. Reloading LifeKeepers list of SCSI disks using the command
`/opt/LifeKeeper/subsys/scsi/actions/lk_cspec`

After that, the resource entry for the volume group needs to be reconfigured. This can be performed through the LifeKeeper GUI. Figure 7-2 shows where to do the reconfiguration.

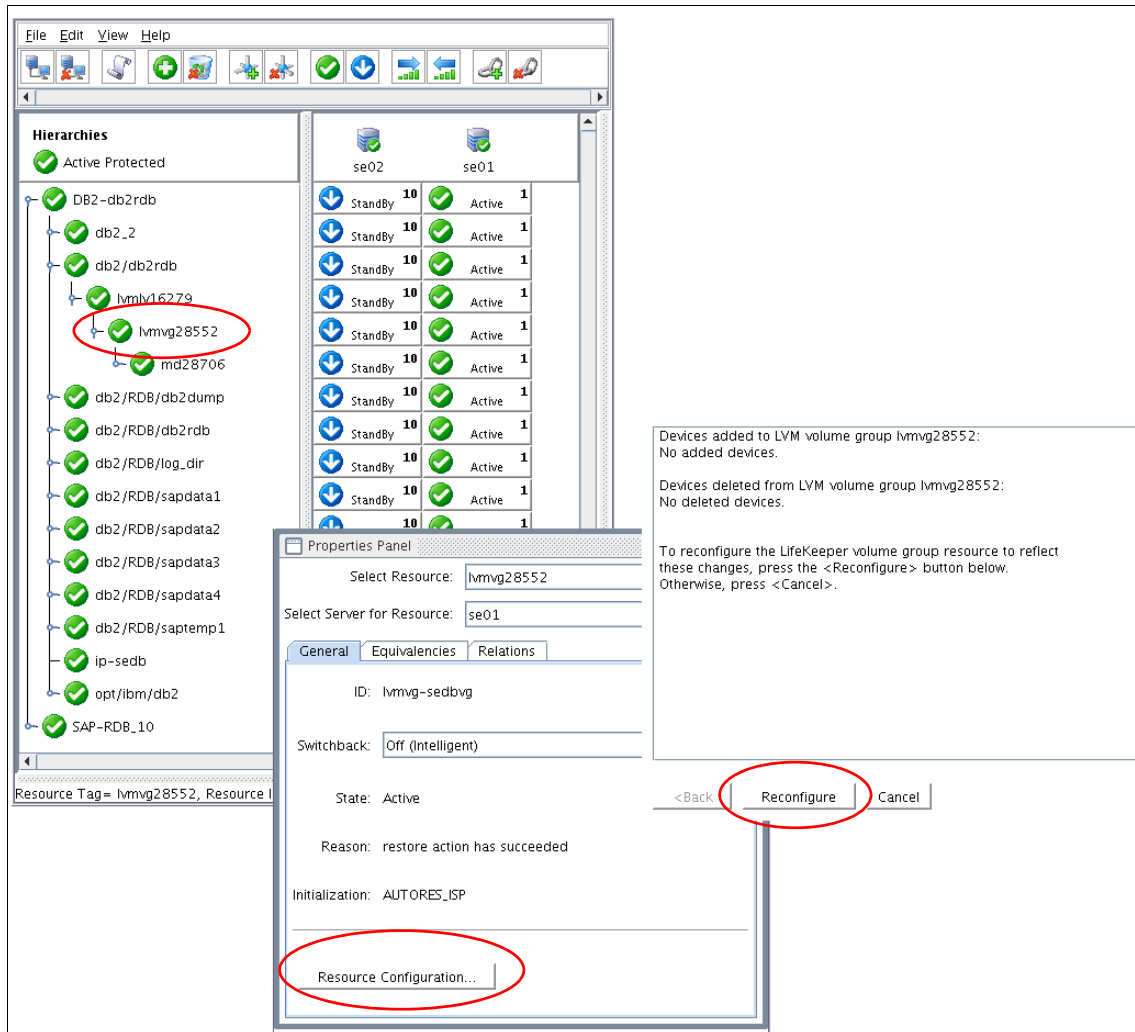


Figure 7-2 Running resource reconfiguration

7.1.5 Frequent file system checks

The ext2/ext3 file systems require a 180-day forced check after the last file system check. Even if a system has been running for that time or the file systems have always been unmounted cleanly, every 180 days the file system check occurs. Depending on the amount of data, files, and the storage speed, this can take a long time. For example, a file system check on ext3 with 100,000 files on a 10 gigabyte file system took about ten minutes on a test scenario. For a terabyte file system, this can take hours.

Turning off the file system checks is a bad idea, because inconsistencies are no longer detected. Therefore, the checks must remain on, and planned file system check maintenance windows are recommended. However, they require a downtime of the application that is running on them.

Together with the Logical Volume Manager snapshot functionality, it is possible to do a semi-online file system check. This check runs on a snapshot logical volume while the file system is still running. Afterwards, the snapshot is tested and if it ends without errors, it is safe to reset the last check date value of the original file system. Otherwise, a full file system check is required.

Creating a snapshot of a Logical Volume Manager based file system is similar to creating a new logical volume. The new volume is a view of a old set of data, while the originating volume can still be changed. The new data is written into the originating volume, and the former data is put temporarily into the snapshot. So the snapshot is even persistent during a reboot.

The minimum required syntax for creating a snapshot volume is:

```
lvcreate --snapshot --size LogicalVolumeSize[kKmMgGtT]}  
--name LogicalVolumeName OriginalLogicalVolumePath
```

The size has to be calculated, depending on the amount of expected changes and the duration of the file system check. For example, if the average write rate is 1 megabyte/second and file system check takes about an hour, the snap shot has to be at least 3.6 gigabytes of space.

Tip: It is best practice to have free space in a volume group available. This can help in a situation where a snapshot is required, or if some file systems run out of space and require quick action.

Example 7-14 shows how a snapshot for a test file system was created with a size of 1 gigabyte.

Example 7-14 Creating a snapshot for a logical volume

```
se02:~ # lvs data00vg  
  LV      VG      Attr  LSize  Origin Snap%  Move Log Copy%  
  installlv data00vg -wi-ao 40.00G  
  testlv   data00vg -wi-ao 10.00G  
se02:~ # lvcreate --snapshot --name testlvsnapshot --size 1G  
/dev/data00vg/testlv  
  Logical volume "testlvsnapshot" created  
se02:~ #
```

The **lv**s command shows the snapshot as a logical volume and displays the usage of the space in the snapshot. It is possible to enlarge the snapshot using the **lvextend** command. If the snapshot runs up to 100%, it becomes unusable.

Example 7-15 shows the snapshot as used after some data has been written to the originating file system.

Example 7-15 Watching snapshot usage

```
se02:~ # lvs data00vg
  LV          VG      Attr   LSize  Origin Snap%  Move Log Copy%
  installlv   data00vg -wi-ao 40.00G
  testlv      data00vg owi-ao 10.00G
  testlvsnapshot data00vg swi-a- 1.00G testlv 50.16
se02:~ #
```

After creating the snapshot, run the **fsck** command on the snapshot logical volume. To make sure it does a full test, use the *force* option as shown in Example 7-16.

Example 7-16 Running a file system check on a snapshot

```
se02:~ # fsck -f /dev/data00vg/testlvsnapshot
fsck 1.38 (30-Jun-2005)
e2fsck 1.38 (30-Jun-2005)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/data00vg/testlvsnapshot: 34939/1310720 files (0.5%
non-contiguous), 1277799/2621440 blocks
se02:~ #
```

After the file system check on the snapshot logical volume has ended successfully and without errors, the snapshot can be removed and the last file system check value of the originating file system can be to the current date using the **tune2fs** command, as shown in Example 7-17.

Example 7-17 Remove snapshot and set current date

```
se02:~ # lvremove /dev/data00vg/testlvsnapshot
Do you really want to remove active logical volume "testlvsnapshot"?
[y/n]: y
Logical volume "testlvsnapshot" successfully removed
se02:~ #
```

```
se02:~ # tune2fs -T 20080311 /dev/data00vg/testlv
tune2fs 1.38 (30-Jun-2005)
Setting time filesystem last checked to Tue Mar 11 00:00:00 2008

se02:~ #
```

Attention: This method does not work correctly with XFS. It requires unmounting the file system before creating the snapshot logical volume and remounting it. This causes a short interruption of the application running on that file system.

XFS uses UUIDs to identify a file system. Attempting to mount that snapshot results in an error message because the snapshot and the originating file system have the same UUIDs. To mount it, the nouuid option needs to be specified.

Other file system types have not been tested in this book.

7.2 SteelEye LifeKeeper administration

This section gives an overview of the basic SteelEye LifeKeeper for Linux administration functionality:

- ▶ “LifeKeeper services”
- ▶ “LifeKeeper Graphical User Interface”
- ▶ “Optional configuration tasks”
- ▶ “Housekeeping”

7.2.1 LifeKeeper services

This section describes how the manual stop and start of LifeKeeper can be performed and how it can be verified.

LifeKeeper start

Typically LifeKeeper is started automatically during the boot sequence of the systems. The LifeKeeper processes are under the control of the init process, which ensures that basic LifeKeeper components are always running.

LifeKeeper can be stopped manually, and if it has been stopped manually, then it has to be brought into service by issuing the following command:

```
/opt/LifeKeeper/bin/lkstart
```


Upon startup, LifeKeeper first tries to establish a communication to other nodes in the cluster. If the instance is the first node in the cluster, it tries to restore the last state of the resource on the node it was running on when it was stopped. For example, if an application was running on that node before, LifeKeeper attempts to start it on that node. If there are other nodes active, then that node joins the cluster. Further actions are dependent on the configuration settings. For example, if the LifeKeeper Switchback type is set to automatic and this node is primary for a resource, a switchback action is initiated.

LifeKeeper stop

The following command stops the LifeKeeper components running on a node. When LifeKeeper is shutting down, all active hierarchies on the node are stopped as well. This behavior can be changed with the use of the switch **-f** or **-r**:

/opt/LifeKeeper/bin/lkstop [-f] [-r] [-n]

If the **-f** switch is used, LifeKeeper only stops itself and no active application hierarchies. With the switch **-r**, LifeKeeper is started automatically on the next reboot.

In the event that the is switch **-n**, LifeKeeper is stopped and all active hierarchies are switched over to the next node.

Note: If LifeKeeper is stopped manually by issuing the command **/opt/LifeKeeper/bin/lkstop [-f] [-n]**, then it can be started during the next reboot. In that case, it has to be started manually.

Detect if LifeKeeper is running

LifeKeeper provides the command **/opt/LifeKeeper/bin/lktest** to detect if its components are running on a node.

The output of this command is empty if LifeKeeper is not running. Otherwise it the output is similar as shown in Example 7-18.

Example 7-18 Main LifeKeeper processes

F	S	UID	PID	PPID	C	CLS	PRI	NI	SZ	STIME	TIME	CMD
4	S	root	5613	1	0	TS	39	-20	1411	Mar07	00:00:00	lcm
4	S	root	5614	1	0	TS	39	-20	1401	Mar07	00:00:00	ttymonlcm
4	S	root	5615	1	0	TS	34	-10	4257	Mar07	00:00:05	lcd

7.2.2 LifeKeeper Graphical User Interface

SteelEye LifeKeeper provides a Graphical User Interface (GUI). Through this user interface, resources can be:

- ▶ Created
- ▶ Deleted
- ▶ Started
- ▶ Stopped
- ▶ Reconfigured
- ▶ Monitored

This section covers these topics:

- ▶ “Graphical User Interface access”
- ▶ “LifeKeeper GUI user management”

Graphical User Interface access

The LifeKeeper GUI is available as a Java applet or a Java application. Before the LifeKeeper GUI can be used, LifeKeeper has to be activated and the GUI Server must be started. In order to start the LifeKeeper GUI Server for the first time, issue the command `/opt/LifeKeeper/bin/lkGUIserver start`.

Subsequently, the GUI Server is started automatically with system reboots if it was not stopped by issuing the `/opt/LifeKeeper/bin/lkGUIserver stop` command. In this case, the start command has to be executed only once. However, if the GUI Server was manually stopped, then it needs to be manually restarted. The Java applet can then be accessed via the uniform resource locator (URL) using an Internet browser as shown in the following example:

`http://<servername>:81`

Note: The `<servername>` value needs to be replaced by your server name.

If there is a firewall between a LifeKeeper GUI Server and the client browser, the network communication probably has to be opened there.

Note: The client must resolve the server name in the short form returned by `/opt/LifeKeeper/bin/sys_list`. This might involve adding domain names to `/etc/resolv.conf` on UNIX or the domain search list on Linux computers.

The GUI as Java application can be reached with command `/opt/LifeKeeper/bin/lkGUIapp` on the LifeKeeper node.

Note: The LifeKeeper GUI can run remotely as an X application tunneled through a Secure Shell X forwarding. This requires:

- ▶ The Secure Shell X forwarding enabled on both the server and the client
- ▶ An X server on the client
- ▶ Some X tools on the server, especially the **xauth** command

Note: VNCserver is also suitable for low bandwidth connections and if the connection is not very stable. Run command **vncserver** on the server and tunnel Port 5900 + Display number of vncserver through SSH (default vnc display=1 --> tunnel 5901:localhost:5901)

If the splash window does not display a **Start** button, or there are other problems during the start of LifeKeeper GUI, refer to the Applet Troubleshooting section. Alternatively, refer to the GUI Network-Related Troubleshooting sections in the LifeKeeper for Linux v6 Planning and Installation Guide, available at:

<http://www.steeleye.com/support>

LifeKeeper GUI user management

The LifeKeeper GUI Server provides a role based authentication mechanism with three classes of GUI users, with different permissions for each:

- ▶ Users with Administrator permission throughout a cluster can perform all possible actions through the GUI.
- ▶ Users with Operator permission on a server can view LifeKeeper configuration and status information, and can bring resources into service and take them out of service on that server.
- ▶ Users with Guest permission on a server can view LifeKeeper configuration and status information on that server.

During installation of the GUI package, an entry for the root login and password is automatically configured in the GUI password file with Administrator permission — thus, allowing root to perform all LifeKeeper tasks on that server via the GUI application or Web client.

Note: The password of the root user is the password during installation time. If you change your root password afterwards, the LifeKeeper “root” password is not changed automatically. You must use **lkpasswd** to change the LifeKeeper root password too.

The recommendation is to allow users other than root access to the LifeKeeper GUI. Especially create users with operator permissions for use by monitoring employees.

User administration is performed through the command line interface, using the **lkpasswd** command.

- ▶ To grant a user Administrator permission for the LifeKeeper GUI, use the following command:
/opt/LifeKeeper/bin/lkpasswd -administrator <user>
- ▶ To grant a user Operator permission for the LifeKeeper GUI, use the following command:
/opt/LifeKeeper/bin/lkpasswd -operator <user>
- ▶ To grant a user Guest permission for the LifeKeeper GUI, use the following command:
/opt/LifeKeeper/bin/lkpasswd -guest <user>
- ▶ To change the password for an existing user without changing their permission level, use the following command:
/opt/LifeKeeper/bin/lkpasswd <user>
- ▶ To prevent an existing user from using the LifeKeeper GUI, use the following command:
/opt/LifeKeeper/bin/lkpasswd -delete <user>

Note: These commands update the GUI password file only on the server being administered. Repeat the command on all servers in the LifeKeeper cluster.

Note: LifeKeeper uses its own name space for users. LifeKeeper user names can be different from operating system users, and the users are not required on the operating system.

The LifeKeeper users can be managed using the **/opt/LifeKeeper/bin/lkpasswd** command.

If the passwords of operating system users are altered, the passwords of LifeKeeper users with the same name are NOT changed.

7.2.3 Optional configuration tasks

This section describes optional possibilities to configure the LifeKeeper for easier handling and using the messaging feature of mdadm for e-mail notification.

Setting environment variables for LifeKeeper

For an easier use of the LifeKeeper commands, it is possible create a profile including the enlarged environment variables PATH and MANPATH for LifeKeeper. Create this profile, for example, in the directory /etc/profile.d and name it LifeKeeper.sh as shown in Example 7-19.

Example 7-19 Profile for LifeKeeper

```
# /etc/profile.d/LifeKeeper.sh for SteelEye LifeKeeper for Linux
PATH=/opt/LifeKeeper/bin:$PATH;export PATH
MANPATH=/opt/LifeKeeper/man:$MANPATH;export MANPATH
```

Integrating mdadm in the LifeKeeper hierarchy

To use the feature of sending messages for events that occur, an additional generic resource needs to be created in the LifeKeeper hierarchy. Typically LifeKeeper stops the mdadm daemon because it is unable to stop an active Software RAID array. Creating of a generic resource is described in 5.8.5, “Creating SAP resources” on page 182.

The following scripts are the simplest way of integrating the mdadm process into a LifeKeeper hierarchy: Example 7-20 shows the Restore script.

Example 7-20 Restore script for mdadm resource

```
#!/bin/bash
/usr/sbin/rcmdadm start
exit 0
```

Example 7-21 shows the quickCheck script.

Example 7-21 QuickCheck script for mdadm resource

```
#!/bin/bash
/usr/sbin/rcmdadm status
exit $?
```

Example 7-23 shows the Recovery script.

Example 7-22 Recovery script for mdadm resource

```
#!/bin/bash
/usr/sbin/rcmdadm restart
exit 0
```

LifeKeeper stops a running mdadm daemon through the Application Recovery Kit providing integration with Software RAID. Given that, the script for removing the mdadm resource in Example 7-23 has no functionality except giving back an exit code of zero.

Example 7-23 Remove script for mdadm resource

```
#!/bin/bash
exit 0
```

7.2.4 Housekeeping

This section covers common administrative tasks and functionality and gives some advice for troubleshooting. It contains these topics:

- ▶ “Status of cluster nodes”
- ▶ “Start of a resource or application”
- ▶ “Stopping any resource or application”
- ▶ “Shutdown of a cluster node”
- ▶ “Shutdown of a complete cluster”
- ▶ “Log files”
- ▶ “Backup of a LifeKeeper configuration”
- ▶ “Listing existing backups”
- ▶ “Restore of the LifeKeeper configuration”
- ▶ “Backup of the LifeKeeper software and licenses”
- ▶ “Collecting useful information in case of failure”

Status of cluster nodes

The LifeKeeper node status can be displayed via the GUI as well as via the command line, as follows:

```
/opt/LifeKeeper/bin/lcdstatus [-d COMPUTER ] [ -q ]
```

This command displays the status of the actual node. It includes the state of the defined resources and hierarchies as well as the state of the heartbeat.

- ▶ If the **-q** option is used, a conclusion of the state is displayed. Normally a detailed output is given.
- ▶ If **-d COMPUTER** option is used, the status of cluster node **COMPUTER** is displayed. Normally the status of the local machine is shown.

The state of every resource shown in Example 7-24 of the **lcdstatus -q** command is interpreted in Example 7-1.

Table 7-1 Status codes for resources

Abbreviation	Resource state	Graphical User Interface
ISP	In Service Protected	Active (green in The LifeKeeper GUI)
ISU	In Service Unimpaired	Active, but in special condition depending on the resource type (for example filesystem > 95% full, yellow in LifeKeeper GUI)
OSU	Out Of Service Unimpaired	Not Active (blue in the LifeKeeper GUI)
OSF	Out of Service Failed	Not Active, Failed (red in LifeKeeper GUI)

Example 7-24 shows the output of the **lcdstatus -q** command:

Example 7-24 Output from the command **lcdstatus -q**

LOCAL	TAG	ID	STATE	PRI0	PRIMARY
se01	SAP-RDB_DVEBMGS12	SAP-RDB_DVEBMGS12	ISP	1	se01
se01	SAP-RDB_10	RDB-10	ISP	1	se01
se01	usr/sap/RDB/ASCS10	/usr/sap/RDB/ASCS10	ISP	1	se01
se01	lvmlv11881	/dev/sesapvg/ascs10lv	ISP	1	se01
se01	lvmsg12642	lvmsg-sesapvg	ISP	1	se01
se01	md12797	eedc5458:a66a1f98:4eeda444:9de774a3	ISP	1	se01
se01	mdComponent12800	md3600507680185000d3800000000000056-1	ISP	1	se01
se01	dmmp12859	3600507680185000d3800000000000056-1	ISP	1	se01
se01	dmmp12964	3600507680185000d3800000000000056	ISP	1	se01
se01	mdComponent13215	md3600507680185000d3800000000000055-1	ISP	1	se01
se01	dmmp13352	3600507680185000d3800000000000055-1	ISP	1	se01
se01	dmmp13497	3600507680185000d3800000000000055	ISP	1	se01
se01	usr/sap/RDB/SCS11	/usr/sap/RDB/SCS11	ISP	1	se01
se01	lvmlv12406	/dev/sesapvg/scs11lv	ISP	1	se01
se01	lvmsg12642	lvmsg-sesapvg	ISP	1	se01

se01	md12797	eedc5458:a66a1f98:4eeda444:9de777a3	ISP	1	se01
se01	mdComponent12800	md3600507680185000d3800000000000056-1	ISP	1	se01
se01	dmmp12859	3600507680185000d3800000000000056-1	ISP	1	se01
se01	dmmp12964	3600507680185000d3800000000000056	ISP	1	se01
se01	mdComponent13215	md3600507680185000d3800000000000055-1	ISP	1	se01
se01	dmmp13352	3600507680185000d3800000000000055-1	ISP	1	se01
se01	dmmp13497	3600507680185000d3800000000000055	ISP	1	se01
...					
se01	db2/RDB/db2rdb	/db2/RDB/db2rdb	OSU	1	se01
se01	lvm1v17755	/dev/sedbvlg/RDB_db2rdb1v	OSU	1	se01
se01	lvmvg28552	lvmvg-sedbvlg	OSU	1	se01
se01	md28706	e996777e:484327ff:cb25cebc:77e7d0df	OSU	1	se01
se01	mdComponent28709	md3600507680185000d3800000000000053-1	OSU	1	se01
se01	dmmp28816	3600507680185000d3800000000000053-1	OSU	1	se01
se01	dmmp28922	3600507680185000d3800000000000053	OSU	1	se01
se01	mdComponent29156	md3600507680185000d3800000000000054-1	OSU	1	se01
se01	dmmp29265	3600507680185000d3800000000000054-1	OSU	1	se01
se01	dmmp29412	3600507680185000d3800000000000054	OSU	1	se01
se01	ip-sedb	IP-9.153.165.99	OSU	1	se01
MACHINE	NETWORK	ADDRESSES/DEVICE	STATE	PRIO	
se02	TCP	192.168.234.1/192.168.234.2	ALIVE	1	
se02	TCP	9.153.165.96/9.153.165.97	ALIVE	2	

Start of a resource or application

A resource can be started both via the command line and the GUI. When starting a resource, LifeKeeper automatically manages the configured dependencies for that resource. If a dependency to a TCP/IP network address or a file system mount is defined, all these necessary resources are started automatically by LifeKeeper before it goes into service. Therefore only a single action has to be issued for starting an application.

Furthermore, LifeKeeper detects autonomously if the resource is active on another node inside the cluster, stops these on the remote node, and starts it, regarding the dependencies.

Note: In order to switch an application or resource from one node to another (switchover), the commands just have to be executed on the target node. LifeKeeper manages all the necessary stoppages on the other nodes.

Graphical User Interface

In order to start a resource using the GUI, choose the resource on the target node on the right hand side, by right mouse click, and select **In Service**. See Figure 7-3.

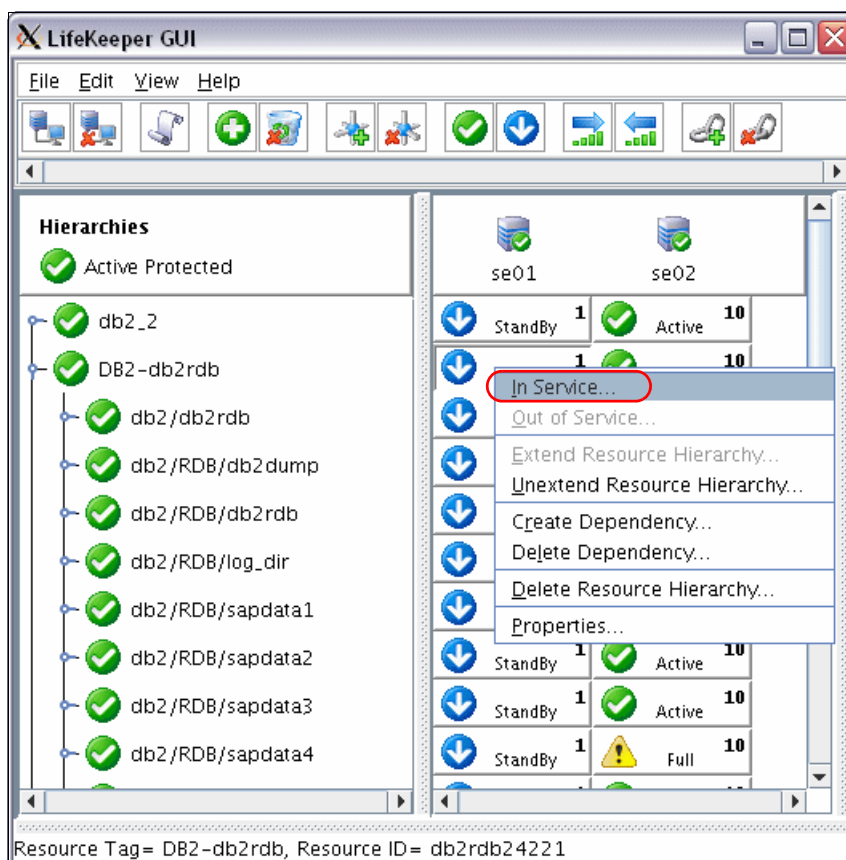


Figure 7-3 In Service

Command line

Command `/opt/LifeKeeper/bin/perform_action -t <TAG> -a restore -b` is executed on the target node:

TAG refers to the resource name used in LifeKeeper. Use the `1cdstatus` command, explained in section “Status of cluster nodes” on page 245 to identify the tag.

The default behavior for the restore action is to bring all objects above and below the specified tag into service. The `-b` option changes this behavior of just the objects below the specified tag.

Stopping any resource or application

An application or resource can also be taken out of service with only a single action, and LifeKeeper automatically removes the configured dependencies for

that resource. All resources depending on the resource being stopped are also shut down before the stop process is executed. Therefore, it is possible to stop the application without being forced to stop TCP/IP network address and file system resources.

If a hierarchy consists of an application and dependent resources (for example: TCP/IP network address and file system) and the TCP/IP network address is stopped, the application is shut down automatically before TCP/IP stops. If the application is stopped, then only this resource is stopped without affecting the TCP/IP network address or file system.

Graphical User Interface

In order to stop a resource via LifeKeeper GUI, choose it by right mouse click and select **Out of Service**. See Figure 7-4.

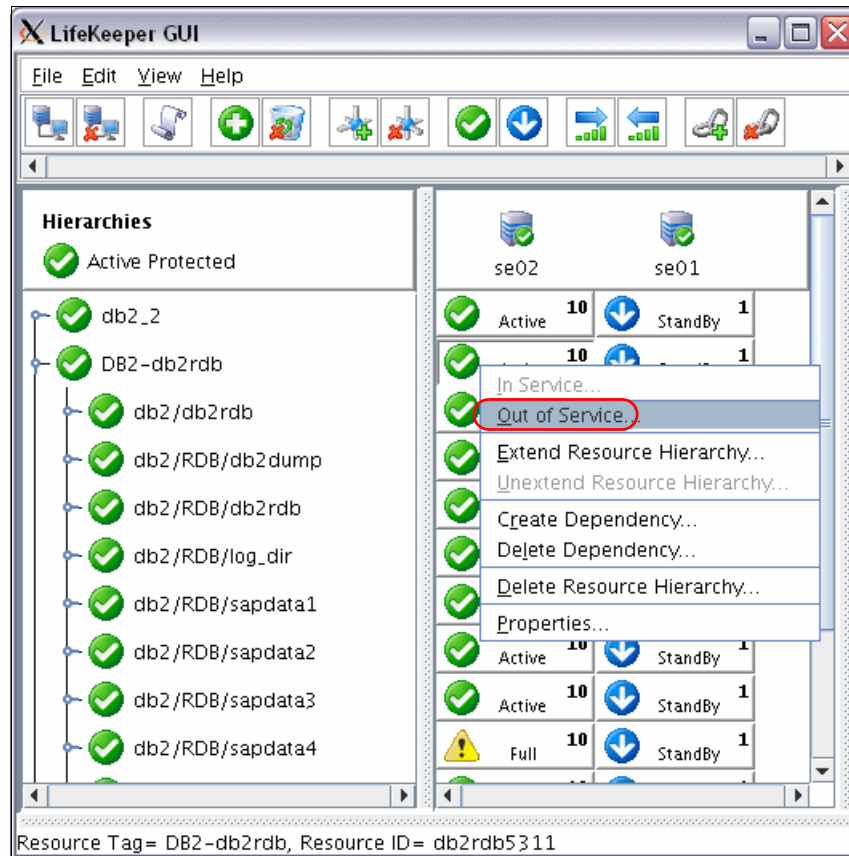


Figure 7-4 Out of Service

Command line

To stop a resource, this command has to be executed on the node where the resource is active now:

```
/opt/LifeKeeper/bin/perfom_action -t <TAG> -a remove
```

TAG refers to the resource name used in LifeKeeper. Use the command `lcdstatus`, explained in “Status of cluster nodes” on page 245 to readout the tag.

Note: Refer to the `perform_action` manpage for further information about the usage of this command.

Shutdown of a cluster node

The behaviour of the cluster while shutting down is based on the defined **Shutdown Strategy**. This can be altered for every node in the GUI via the Properties dialogue by right-clicking on the server icon. See Figure 7-5.

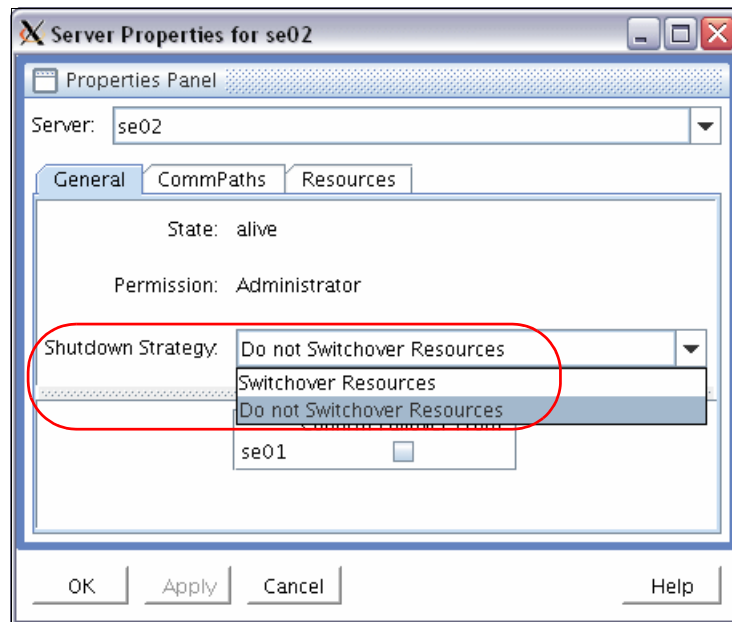


Figure 7-5 Server Properties Panel

► **Switchover Resources:**

If the node shuts down via **init 6**, **init 0** or **reboot**, LifeKeeper stops all active resources and itself afterwards. All active resources of the node being shut down are taken over by one of the remaining nodes.

► **Do Not Switchover Resources (default):**

If the node shuts down via **init 6**, **init 0** or **reboot**, LifeKeeper stops all active resources and itself afterwards. All active resources of the node being shut down are *NOT* taken over by one of the remaining nodes.

Note: **lkstop** is no operating system shutdown, only a system shutdown with **init 6**, **reboot** or **halt** is covered by this command.

Shutdown of a complete cluster

On a shared storage cluster both nodes can be stopped in any order. While the cluster is starting, the same state is tuned in as when the system was shut down. Resources are activated on the same node as they were before.

Resources that have not been active would not be activated either.

Log files

To analyze failover situations or error situations, it is helpful to use the LifeKeeper Log Files. The Log Files can be shown both via the LifeKeeper GUI and the command line.

Graphical User Interface

To show the LifeKeeper Log Files via the LifeKeeper GUI, choose the server by right mouse click and select **View Logs**. See Figure 7-6.

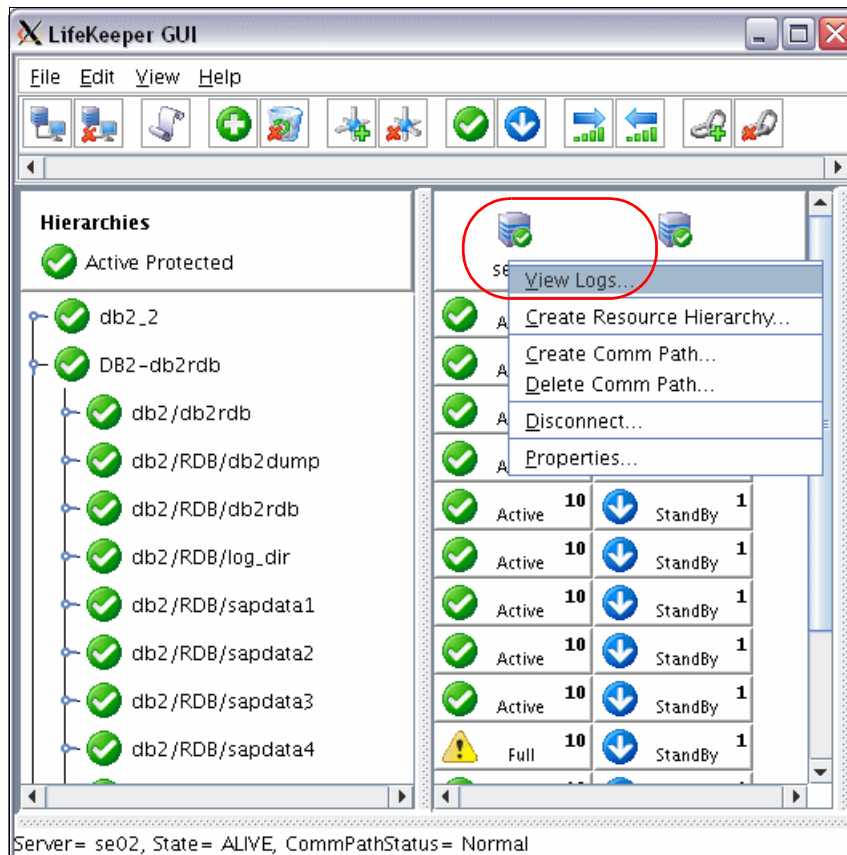


Figure 7-6 Choose View Logs

The next window shows by default the updates of the LifeKeeper log. This log contains information regarding the management of LifeKeeper protected applications and LifeKeeper resources. Additionally, information generated by the application's remove and restore scripts is stored in this log. Finally, major LifeKeeper events such as stopping and starting LifeKeeper, service and failover operations, and resource health monitoring activities are recorded in this log.

Another type of log files and the length of displayed log file can be selected. See Figure 7-7.

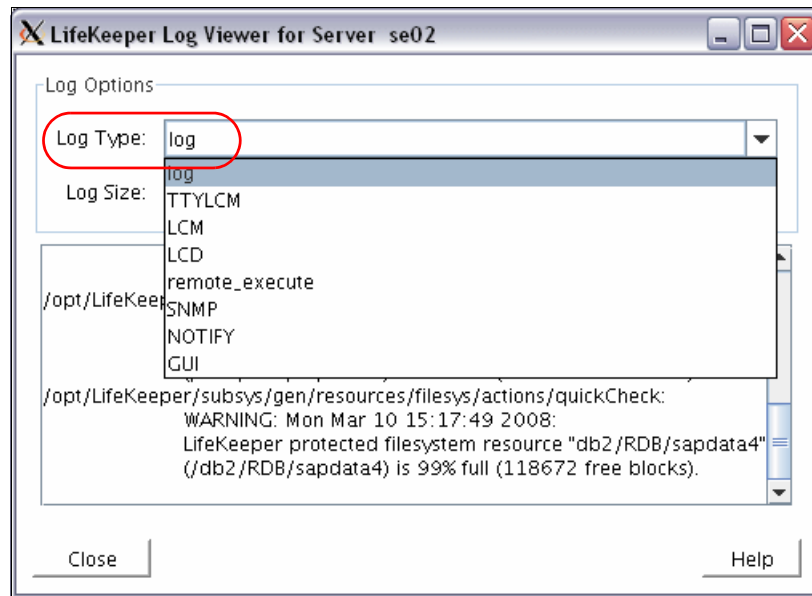


Figure 7-7 Log file types

Further information about the log file types can be found in the **1k_log** manual page (available through the command **man 1k_log** on a server with LifeKeeper installed).

Command line

At the command line, the log files can be displayed by executing the following command:

```
/opt/LifeKeeper/bin/1k_log [-t NUMBER] [ -f ] log
```

Besides this log, additional LifeKeeper logs are available.

- ▶ If **-f** is used, LifeKeeper follows the log-display (such as **tail -f**).
- ▶ If **-t NUMBER** is used, only the *number* of last lines is shown. Without **-t**, the whole log is displayed.

Note: The LifeKeeper logs are cyclic logs, which means that they have a configurable maximum size (in `/etc/default/LifeKeeper`). For this reason, an overflow of the log file is not possible.

Backup of a LifeKeeper configuration

A backup of the whole LifeKeeper resource configuration can be achieved using the following command:

```
/opt/LifeKeeper/bin/lkbackup -c [--cluster]
```

If **--cluster** is added, the command **lkbackup** is executed in the whole cluster, thus a backup is made on all cluster nodes. For the connection to the other cluster nodes, the **lkbackup** command is using the **ssh** command.

The archives are gzip compressed tar archives (.tar.gz) and they are deployed to the /opt/LifeKeeper/config directory. The naming schema is:

```
archive.YYMMDDHHMM.tar.gz
```

The size of the archive is about 15 kilobytes only. We recommend creating a backup of the LifeKeeper configuration before any changes at the LifeKeeper configuration are committed.

Note: The backup of the LifeKeeper configuration does not replace a backup of the LifeKeeper nodes; **lkbackup** does not save programs or licences!

Listing existing backups

This command lists all available LifeKeeper backups:

```
/opt/LifeKeeper/bin/lkbackup -l [--cluster]
```

Restore of the LifeKeeper configuration

If a backup configuration has to be restored, the LifeKeeper *must not be active* at this time. After LifeKeeper has been stopped, the archived backup configuration can be applied with the following command:

```
/opt/LifeKeeper/bin/lkbackup -x [-f ARCHIVE ] [--cluster]
```

Where **-f ARCHIVE** specifies the archive file. If this option is not specified, then the latest archive found in the default location is used.

Note: Refer to the **lkbackup** manual page for additional information about possible options.

Backup of the LifeKeeper software and licenses

In order to back up LifeKeeper, the following directories have to be saved:

- ▶ /opt/LifeKeeper
- ▶ /etc/init.d/*lifekeeper
- ▶ /etc/default/LifeKeeper
- ▶ /var/LifeKeeper

The following command creates a gzipped tar archive of all necessary data:

```
tar -czvf /root/LK_BACKUP-$(hostname)-$(date -I).tgz \  
/opt/LifeKeeper /var/LifeKeeper /etc/default/LifeKeeper \  
$(find /etc -name \ “*lifekeeper*”)
```

A backup created by this method includes software as well as information about the configuration.

Note: If LifeKeeper is restored without installing RPM's first, the RPM database does not contain the package information about LifeKeeper.

Collecting useful information in case of failure

To collect important system information in case of failure, LifeKeeper provides the command **lksupport**. This command collects the system information under the directory /tmp/lksupport/<hostname> and creates a gzipped tar archive named <hostname>.lksupport.<time stamp>.tar.gz.

Example 7-25 shows the output of a running **lksupport** command.

Example 7-25 Output lksupport

```
se02:~ # lksupport  
Collecting info under /tmp/lksupport/se02  
  saving LifeKeeper status  
  saving LifeKeeper logs  
  saving LifeKeeper defaults file  
  saving md and SDR data  
  saving drbd data  
  saving LifeKeeper device_info files  
  saving LifeKeeper SCSI kit files  
  saving LifeKeeper configuration information  
  saving host information  
  saving LifeKeeper licensing data  
  saving network data  
  saving installed package data  
  saving process information
```



```
saving LVM data (this may take minutes to complete)
saving /proc data
saving module configuration data
saving lsmod data
saving file system information
saving boot loader data
saving system timestamps
saving Device Mapper information
saving Device Mapper - multipath information
saving system information
saving NFS data
```

```
Creating support file /tmp/lksupport/se02.lksupport.0803121457.tar.gz
```

7.3 Customizing of LifeKeeper parameters

A LifeKeeper installation can be customized to individual requirements. This section covers these topics:

- ▶ “Changing global operational parameters”
- ▶ “Changing LifeKeeper configuration values”

7.3.1 Changing global operational parameters

LifeKeeper allows customizing of many operational parameters. These are all stored in a configuration file called `/etc/default/LifeKeeper`. In this section we show some examples:

- ▶ The *LKCHECKINTERVAL* variable contains the interval for application health checks. The default is 120 seconds. If faster reactions are required, a lower value should be used. LifeKeeper has to be restarted to apply a change of this value.
- ▶ The *LK_NOTIFY_ALIAS* value contains an e-mail address that receives notifications about cluster actions such as quick check errors, recovery, start, and stop of LifeKeeper itself and cluster resources.
- ▶ The *LK_TRAP_MGR* can contain the host name or network address of a SNMP trap receiver. LifeKeeper sends SNMP traps for cluster actions.
- ▶ The *LOGFILE* values contain the maximum size of the different LifeKeeper log files. LifeKeeper has to be restarted to apply a change of this value.

- ▶ The *FILESYSFULLWARN* and *FILESYSFULLERROR* values contain threshold values for warning and error log messages for protected shared storage. Messages regarding the error threshold are sent through e-mail if configured.
- ▶ The *LCMHBEATTIME* specifies the interval of heartbeat checks and *LCMNUMHBEATS* specifies the number of heartbeats allowed to fail before a communication path is considered dead.

Changing this parameter allows increasing or decreasing tolerance against network outages. LifeKeeper has to be restarted to apply a change of these values.

Attention: All changes to the general parameter file must be made very carefully because they can cause unexpected behavior and even dysfunction of the LifeKeeper software. The file must be equal on all nodes of a cluster.

7.3.2 Changing LifeKeeper configuration values

There are a number of values in LifeKeeper that might need to be changed after LifeKeeper has been configured and set up. Examples of values that can be modified include the uname of LifeKeeper servers, communication path TCP/IP addresses, TCP/IP resource addresses, and tag names. To change these values, carefully follow these instructions:

1. To get a point of return in the event that the next steps generate a faulty or unintended configuration, first create a backup configuration of the LifeKeeper resources by issuing the following command:
2. Stop LifeKeeper on all servers in the cluster by using the command:
3. To change a value of the LifeKeeper configuration which is associated with the changes of the operating system configuration, carry out the operating system configuration at first.
4. First, verify that the changes to be made do not have any unexpected results by examining the output of running this command:

```
/opt/LifeKeeper/bin/lkbackup -c --cluster
```

```
/opt/LifeKeeper/bin/lkstop
```

```
/opt/LifeKeeper/bin/lk_chg_value -Mvo <old_value> -n <new_value>
```

The **-M** option specifies that no modifications should be made to any LifeKeeper files.

If more than one LifeKeeper value is to be changed, old and new values must be specified in a file on each server in the cluster in the following format:

```
old_value1=new_value1
old_value2=new_value2
....
```

To use this file to change the values, use the command:

```
/opt/LifeKeeper/bin/lk_chg_value -Mvf file_name
```

Verify the output of the command.

5. If the command verification is as expected, then modify the LifeKeeper files by running the command **lk_chg_value** on all cluster servers:

```
/opt/LifeKeeper/bin/lk_chg_value -vo <old_value> -n <new_value>
```

or

```
/opt/LifeKeeper/bin/lk_chg_value -vf file_name
```

6. Restart LifeKeeper on all servers by using the command:

```
/opt/LifeKeeper/bin/lkstart
```

If the cluster is being viewed using the LifeKeeper GUI, it might be necessary to close and restart the GUI.

Further information on the command **lk_chg_value is available** is available from the manpage. Additional information and examples are also in the Online Help accessible at the following URL:

<http://<servername>:81/help/lkstart.htm>

Note: The <servername> has be replaced with one of your cluster servers.

7.4 Maintenance during uptime

It is important to ensure that the system or protected applications are maintained by LifeKeeper and all applications are in service within the cluster node.

If an application is stopped by an administrator outside of LifeKeeper, then LifeKeeper starts this application after the next faulty QuickCheck again. Therefore, if an application has to be maintained, use the LifeKeeper GUI or use the command line **perform_action** interface. After this, the administrator is able to maintain the application, including start and stop, without LifeKeeper effects.

Making a resource Out of Service is described in “Stopping any resource or application” on page 247.

An application can be brought back In Service by using the LifeKeeper GUI or by using the command line **perform_action** interface.

Bringing a resource In Service is described in “Start of a resource or application” on page 246.

Another way to end resource monitoring is by stopping LifeKeeper with the following command:

```
/opt/LifeKeeper/bin/lkstop -f
```

The resources remain running on the local system, but are no longer protected by LifeKeeper. This option should be used with caution, because if resources are not gracefully shut down, then items such as SCSI locks are not removed. If the system on which **lkstop -f** is executed subsequently fails or is shut down, then the system does NOT initiate failover of the appropriate resources. After reboot, resources that remained running would no longer be running on any system in the LifeKeeper cluster.

After the maintenance session completes successfully, then start LifeKeeper by running the following command:

```
/opt/LifeKeeper/bin/lkstart
```

The third way to end monitoring of one or more resources is by running the following command, where **tag** is the LifeKeeper name of the resource:

```
/opt/LifeKeeper/bin/ins_setstate -t <tag> -S OSU
```

For example, the monitoring of the database resource can be stopped by running the command:

```
/opt/LifeKeeper/bin/ins_setstate -t DB2-db2rdb -S OSU
```

The current state of the resource is shown in Figure 7-8.

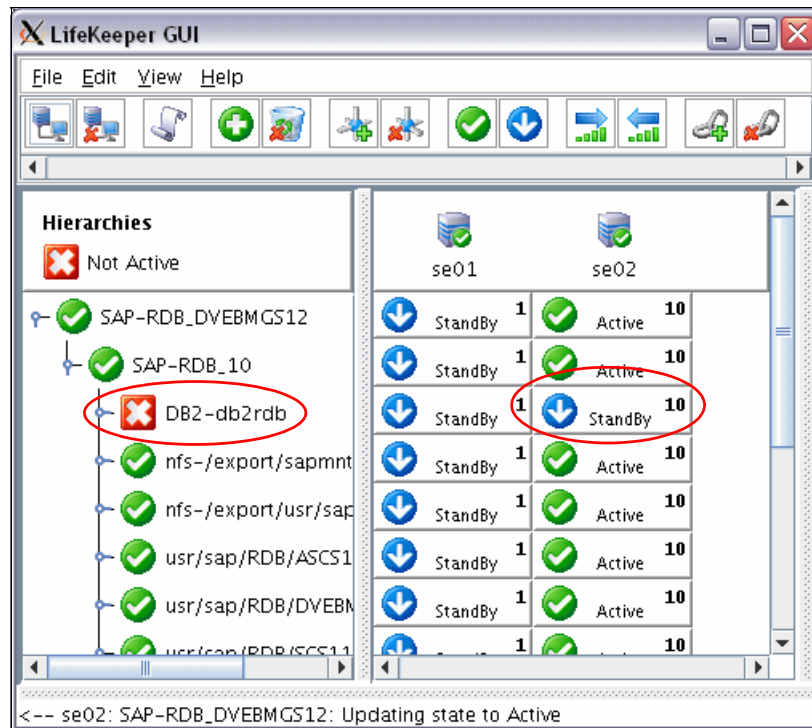


Figure 7-8 LifeKeeper GUI with one resource Out of Service inside the hierarchy

After a successful maintenance task, bring back the resource to monitoring by LifeKeeper by running the command:

```
/opt/LifeKeeper/bin/ins_setstate -t DB2-db2rdb -S ISP
```

This command executes a check if the application is running; if not, LifeKeeper starts the application.

7.5 Backup and restore

Backup and restore processes in a clustered environment require some special attention. Application data can usually be accessed only from one node exclusively at a time. According to that, backup of the data has to be run from the node where the data is accessible. The backup data on a typical backup server such as IBM Tivoli Storage Manager is bound to a node name or the host name of the data source.

Therefore, a backup solution for a cluster service must be performed so that correct saved data is located when required. This section provides a conceptual solution based on best practice. In this solution, the cluster service backup data is stored under a common name for that data. This common name is independent from the physical node on which the cluster service is running. Data is backed up under this common name from the node that runs the service actively during backup.

With the IBM Tivoli Storage Manager backup and archive client, it is possible to have multiple logical configurations on one physical machine. These configurations are registered in the IBM Tivoli Storage Manager server as nodes. Each configuration can back up different data sets. One data set is created for the physical node's local file systems on internal disks. Further configurations with individual node names are created for each cluster service, for example, one for **sesap** and one for **sedb**. Multiple copies of the IBM Tivoli Storage Manager client software, especially the scheduler process (dsmsched) can be started on one node. One instance is started through the cluster software, related to a cluster service.

This can be achieved with SteelEye LifeKeeper by creating start and stop scripts for the IBM Tivoli Storage Manager scheduler process and integrating them into the cluster using the Generic Application Recovery Kit. With a dependency to the service TCP/IP network address, it is started, stopped and moved with the application.

Figure 7-9 shows an example with two backup clients in the cluster hierarchy. There is one for each service, sesap and sedb.

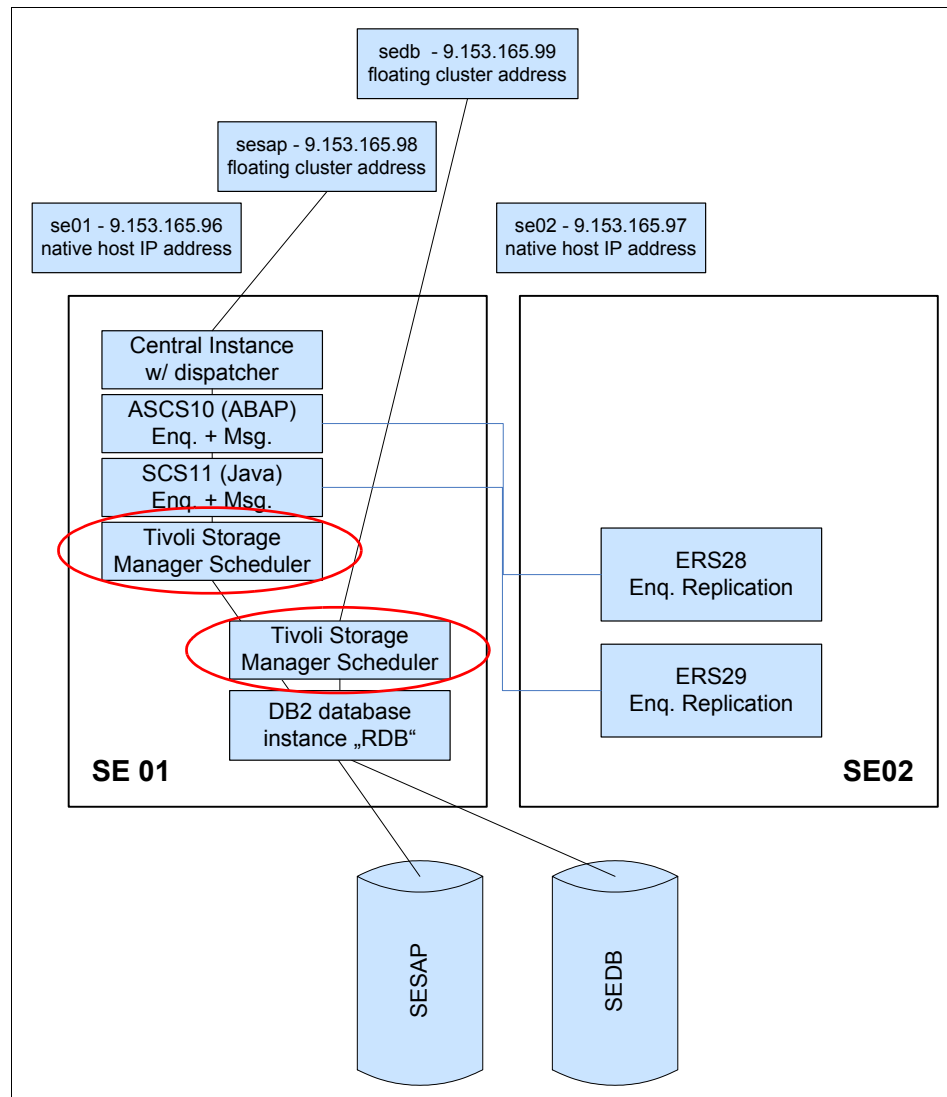


Figure 7-9 Adding backup clients to the cluster hierarchy



Troubleshooting

This chapter provides information pertaining to issues encountered during the high availability installation. It also covers some additional tips and techniques that can be helpful in problem determination.

We discuss the following topics:

- ▶ “Installation”
- ▶ “DB2”
- ▶ “SAP NetWeaver”
- ▶ “LifeKeeper”

8.1 Installation

In this section we consider some issues encountered during installation.

SAPinst DB2 instance creation

The SAP GUI installer encountered a problem during the task which creates the DB2 LUW 9.5 instance. The parameters passed to the **db2icrt** command were incorrect. The SAP GUI installer had included the parameter for the word width, which has been discontinued in DB2 9.5.

DB2 9.5 provides support for both 32-bit and 64-bit installations for Linux on x86, however the 64-bit version is recommended. To review the current requirements visit:

<http://www-306.ibm.com/software/data/db2/9/sysreqs.html>

Although both word widths are supported, the word width is now determined by the operating system. In DB2 Version 9.1, this option returned a warning message, but in DB2 Version 9.5 it returns a syntax error.

To review the changes implemented concerning the word width parameter, visit:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.qb.migration.doc/doc/c0022266.html>

Example 8-1 is an extract from the sapinst.log regarding the **db2icrt** error.

Example 8-1 Output from sapinst.log

ERROR 2008-02-22 18:52:53

```
FC0-00011 The step CreateInstance with step key
|NW_Doublestack_DB|ind|ind|ind|ind|0|0|NW_CreateDBandLoad|ind|ind|ind|i
nd|9|0|NW_CreateDB|ind|ind|ind|ind|0|0|NW_DB6_DB|ind|ind|ind|ind|1|0|NW
_DB6_CreateDB2Instance|ind|ind|ind|ind|2|0|CreateInstance was executed
with status ERROR.
```

To work around this issue, the instance can be created manually and the SAP GUI installation restarted. The following command was issued to create the instance manually in the environment for this book:

```
# /opt/ibm/db2/V9.5/instance/db2icrt -a SERVER_ENCRYPT -u db2rdb db2rdb
```

The same technique can be used to install the client. For the client instance creation, use the **-s client** parameter to indicate the type of instance. By default the **db2icrt** command uses **ese** indicating a DB2 Enterprise Server Edition.

LifeKeeper and Device-Mapper MultiPath

During integration of the shared storage into the SteelEye LifeKeeper cluster hierarchy, an error occurred showing that LifeKeeper was not able to find the disks that have been given the alias name `/dev/disk/by-name/sesap0`.

From the first release of SUSE Linux Enterprise Server 10 to the Service Pack 1, the behavior of the Device-Mapper Multi-Path I/O changed slightly. The names are built either by using the world-wide name of a Storage Area Network volume or by using an alias configured in the `/etc/multipath.conf` configuration file.

For the test scenario, the names for the Storage Area Network volumes selected were:

- ▶ `sedb0`
- ▶ `sedb1`
- ▶ `sesap0`
- ▶ `sesap1`

As soon as a partition was created on these volumes, they got a suffix like `p0`, `p1`, `p2`, and so on, to show the number of the partition. Files that have such a suffix are ignored by the SteelEye Lifekeeper Application Recovery Kit for Device-Mapper Multipath. LifeKeeper interprets these as partitions. So the multi-path connected disks were inadvertently ignored. To work around this issue, the names were changed to:

- ▶ `sesap-0`
- ▶ `sesap-1`

LifeKeeper then recognized the devices with the new names without difficulties.

Starting with Service Pack 1, the SUSE Linux Enterprise Server 10 uses another suffix for partitions in multipath disks. The suffix is `-part1`, `-part2`, `-part3`, and so on.

Tip: The test scenario was built with the Application Recovery Kit for Device-Mapper Multipath *steeleye-lkDMMP-6.2.1-2*. With this version of the Application Recovery Kit, avoid alias names for multipath devices ending in `p0`, `p1`, and so on.

8.2 DB2

In this section we discuss issues encountered with the database:

- ▶ Troubleshooting a problem with the `db2nodes.cfg` file
- ▶ Tools to aid in diagnosis

Troubleshooting a problem with the `db2nodes.cfg` file

Since DB2 V8, single and partitioned instances have a `db2nodes.cfg` file in the `sqllib` subdirectory. In a clustered environment, if the instance home directory is placed on shared disk, then consideration must be taken to ensure that the entry is correct for failover and failback scenarios. If the `db2nodes.cfg` entry is incorrect, the **`db2start`** command fails with an `SQL6031N` or a similar error, indicating trouble accessing the `sqllib` directory.

To avoid this issue, LifeKeeper manages the `db2nodes.cfg` file using the **`db2gcf -u`** command option.

Tools to aid in diagnosis

A key tool available in troubleshooting db2 is the **`db2diag`** command. The `db2diag.log` is the primary administrative log, and the **`db2diag`** command is a tool that helps filter and format the volume of information found in the `db2diag.log`.

Use the following command, for example, to quickly review any severe level errors that have been encountered:

```
db2diag -g level=severe
```

To see only the severe level errors from the last three days, issue the following command:

```
db2diag -gi "level=severe" -H 3d
```

It is possible to traverse the `db2diag.log` output using other tools such as `awk` or `grep`. The `db2diag` location is determined by the `DIAGPATH` configuration parameter of the database manager (`dbm`). The default location is `$HOME/sqllib/db2dump`.

DB2 has the ability to install multiple copies of DB2 products on the system with the flexibility to install these products in a desired path.

To aid in tracking where each product is installed, the **`db2ls`** command lists the DB2 products and features on the system.

Example 8-2 Sample listing of DB2 products and features

```
# db2ls
```

Install Path	Level	Fix Pack	Special	Install Number	Install Date	Installer UID
/opt/ibm/db2/V9.5	9.5.0.0	0		1	Fri Feb 22 18:26:45 2008 CET	0

To search the DB2 Technical Support Web site for additional information and problem resolution, refer to:

<http://www.ibm.com/software/data/db2/udb/support>

If it is necessary to involve DB2 Technical Support in problem resolution, DB2 Technical Support might require information captured from the db2support utility. This utility gathers information on both the operating system and database levels.

An example of invoking the **db2support** command is as follows:

```
db2support /tmp -d RDB -c -s
```

This collects the information and store the zip file in the /tmp directory using detailed collection (-s).

Detailed information on troubleshooting tools for DB2 can be found in the DB2 information library located at:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp>

8.3 SAP NetWeaver

On occasion there are problems that can occur and prevent the SAP server processes from starting properly. The best place to start analyzing the cause of these problems is by reviewing the log and trace files.

For the SAP Java application server, these are located in the work and log directory structures.

In the work directory, the following file provides key information for troubleshooting:

- dev_jcontrol

Also, consider these files for each process that can be bootstrap, dispatcher, or server<n>:

- ▶ dev_<processname>
- ▶ jvm_<processname>
- ▶ std_<processname>

For Java processes, dispatcher and server node information can be found under the /cluster subdirectory.

Figure 8-1 depicts a hierarchy view of key log and trace files for problem determination.

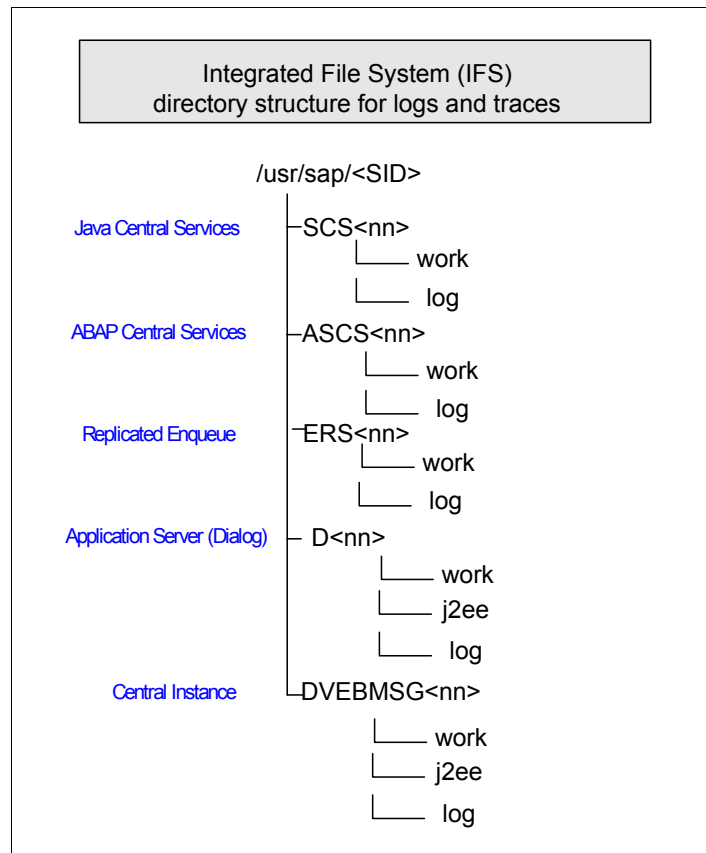


Figure 8-1 Integrated File System (IFS) directory structure

Additional troubleshooting information is available through the SAP installation guides available at:

<https://service.sap.com/instguides>

8.4 LifeKeeper

The **lk_log** command can be used to monitor activity and display information from the log. The LifeKeeper commands are located in `/opt/LifeKeeper/bin` by default.

Here is an example command to review the LifeKeeper log:

```
#/opt/LifeKeeper/bin/lk_log log -t 20 -f
```

The **lcdstatus** command can be used to review the current state of resources. Here is an example command using **lcdstatus**:

```
# /opt/LifeKeeper/bin/lcdstatus
```

For more information relating to command line features for LifeKeeper, refer to the Command Line Interface Guide:

<http://licensing.steeleye.com/documentation/linux.html>

Surpressing terminal type tset errors in the LifeKeeper log

SAP and the database recovery kits output numerous tset errors to the LifeKeeper log. These are a result of using the **su** commands in a non-interactive shell. Example 8-3 shows some typical text from the LifeKeeper log.

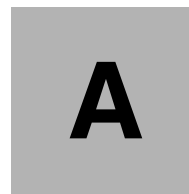
Example 8-3 output from lk_log

```
tset: standard error: Invalid argument
```

The recommended workaround is documented in detail in the LifeKeeper for Linux SAP Recovery kit manual. Refer to the following Web site:

<http://licensing.steeleye.com/support/docm.php>

There are specific modifications required to adjust the terminal connection. The silent parameter of the **ty** is implemented for both the **c** and **bash** shell.



Additional material

This book refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser at:

<ftp://www.redbooks.ibm.com/redbooks/SG247537>

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG247537.

Abbreviations and acronyms

ACR	Automatic Client Re-route	LUW	Linux, UNIX, and Windows
ARKs	Application Recovery Kits	LUW	logical unit of work
ARP	Address Resolution Protocol	MDM	Master Data Management
AS	Application Server	NAS	Network Attached Storage
ASCS	ABAP System Central Services	NFS	Network File System
ATS	America's Advanced Technical Support	NIC	network cards
BI	Business Intelligence	OSI	Open Systems Interconnection
CI	Central Instance	RAID	redundant array of independent disks
CIFS	Common Internet File System	RPM	RedHat Package Manager
DAS	DB2 Administration Server	SAN	Storage Area Network
DAS	Direct Attached Storage	SCS	System Central Services
DMMP	Device Mapper Multipath	SDD	Subsystem Device Driver
DNS	Domain Name Server	SDM	software deployment manager
DPF	DB2 data partitioning feature	SPOFs	single points of failure
DRBD	distributed replicated block device	STG	Systems and Technology Group
EP	Enterprise Portal	SVC	SAN volume controllers
ESE	Enterprise Server Edition	UPS	uninterruptable power supplies
GUI	Graphical User Interface	URL	uniform resource locator
HADR	High Availability Disaster Recovery	iSCSI	Internet Small Computers System Interface
HBA	host adapter		
IBM	International Business Machines Corporation		
IDS	into Informix Data Server		
IFS	Integrated File System		
ISICC	IBM SAP International Competence Center		
IT	Information technology		
ITSO	International Technical Support Organization		
JCE	Java Cryptography Extension		

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 278. Note that some of the documents referenced here may be available in softcopy only.

- ▶ IBM System Storage DS8000: Copy Services in Open Environments
<http://www.redbooks.ibm.com/abstracts/sg246788.html>
- ▶ IBM System Storage DS6000™ Series: Architecture and Implementation
<http://www.redbooks.ibm.com/abstracts/sg246781.html>
- ▶ IBM System Storage DS4000 and Storage Manager V10.10
<http://www.redbooks.ibm.com/abstracts/sg247010.html>
- ▶ High Availability and Scalability Guide for DB2 on Linux, UNIX®, and Windows
<http://www.redbooks.ibm.com/abstracts/sg247363.html>
- ▶ DB2 9.1 Data Recovery and High Availability Guide and Reference
<http://www-1.ibm.com/support/docview.wss?uid=pub1sc10422800>
- ▶ IBM System x and BladeCenters
<http://www.redbooks.ibm.com/abstracts/redp4234.html>

Online resources

These Web sites are also relevant as further information sources:

- ▶ SAP Web Application Server support matrix
- ▶ SAP Web Application Server
- ▶ SAP and Novell mutual support for SAP applications
<http://www.novell.com/products/server/sap.html>

- ▶ Sizing SAP on IBM Systems
<http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS2336>
- ▶ Definition of high availability clusters
http://en.wikipedia.org/wiki/High-availability_cluster
- ▶ SteelEye LifeKeeper software certification details
<http://www.SteelEye.com/products/linux/>
- ▶ IBM hardware supported on the Novell SUSE Linux Enterprise Server
<http://developer.novell.com/yesssearch/Search.jsp>
- ▶ LifeKeeper Data Replication
<http://www.SteelEye.com/products/datarep.html>
- ▶ Configuring SAP with DB2 HADR on developerWorks
<http://www.ibm.com/developerworks/db2/library/techarticle/dm-0508zeng/>
- ▶ High availability techniques for DB2
<http://www-1.ibm.com/support/docview.wss?uid=publsc10422800>
- ▶ SAP NetWeaver® 2004s product documentation
<http://service.sap.com/nw2004s>
- ▶ SteelEye LifeKeeper support on several IBM System p and x servers
<http://www.SteelEye.com/products/linux/>
- ▶ NetWeaver Technical Infrastructure
<https://service.sap.com/installnw7>
- ▶ IBM Insight for SAP and the related documentation
<http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS381>
- ▶ LifeKeeper 6.2.1 on a Linux SUSE Linux Enterprise Server 10 for SAP NetWeaver
<http://www.steeleye.com/support>
- ▶ Certified Linux distributions for DB2
<http://www.ibm.com/software/data/db2/linux/validate>
- ▶ DB2 hardware and software prerequisites
<http://www.ibm.com/software/data/db2/udb/sysreqs.html>
- ▶ Java Cryptography Extension (JCE) policy
<http://www6.software.ibm.com/dl/jcesdk/jcesdk-p>

- ▶ setting up the enqueue replication server
http://help.sap.com/saphelp_nw2004s/helpdata/en/de/cf853f11ed0617e1000000a114084/content.htm
- ▶ SteelEye LifeKeeper license keys
<http://www.steel-eye.com/support>
- ▶ Relocating the LifeKeeper package
<http://www.steel-eye.com/support>
- ▶ Backup Interface and Local Recovery Configuration Restrictions
<http://www.steel-eye.com/service>
- ▶ SAP Marketplace:

Note: The SAP Marketplace is a secure site and requires a user ID and password.

- SAP Note 766222: Information on supported IBM hardware for Linux
<http://service.sap.com/notes>
- SAP Note 1089578: Using DB2 9.5 with SAP software
- SAP Note 919550: DB6 SAP NetWeaver 2004s installation
- SAP Note 101809: Supported Fix Packs for DB2 and database software download instructions
- SAP Note 958253, SAP Note 95825: SAP installation documentation for specific hints for the Linux base operating system installation
- SAP Note 768727: Event of failure of one of the two components
<http://service.sap.com/notes>
- Installation of the SAP NetWeaver 7.0
<http://service.sap.com/instguidesNW70>
- installations guides
<http://service.sap.com/instguides>

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional Materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

Numerics

9s of availability 3

A

ABAP Central Services 14
ABAP central services 83
ABAP clusters 46
ABAP development 11
ABAP enqueue servers 81
ABAP replication server 201
ABAP stacks 143
ABAP system 200
ABAP System Central Services 46
ABAP+Java AddIn environments 183
active hierarchies 239
Active Slave 225
active/active 74
active/active configuration 54, 194
active/passive 74
active/passive configuration 53, 194
active-backup configuration 66
Address Resolution Protocol 30
administration considerations 2
administrative tasks 223
Administrator permission 241
Advanced Technical Support 91
analysis process 91
application availability 4
application clusters 21
application instance failure 215
Application layer 12
Application Platform 10
Application processes 12
application recovery 16
Application Recovery Kits 16–17, 24
Application Server 50, 83
application specific 17
application threads 51
architectural considerations 2
architectural decisions 25, 58
ARP-based monitoring 66
Array based replication 38
ASCS10 instance 83

asynchronous data 37
asynchronous replication 41
authentication mechanism 241
Automated cluster management 24
automated failure detection 19
Automatic Client Re-route 40
automatic startup of RAID 69
Automatic switchback 167
automatic system recovery 16
AVA Central Services 14
availability 20
availability implementations 2
availability requirements 20
availability topologies 2
avoid unplanned outages 6

B

backup 253
backup clients 261
backup interface 66, 174
balance workload 47
base operating system 2, 99
base software stack 23
Best practice methods 2
Blade Center 63
Blade Center Management Module 63
blade server technology 21
Blade servers 60, 63
bonding feature 66
broadcast messages 32
broadcast packet 31
build a cluster 17
built-in high availability 37
Business Intelligence 12
business operations 19
business requirement 19

C

Capacity 20
capacity level 91
cascading failure 52
Central instance 83
Central Instance components 85

- central instance failure 215
- central processing unit 23
- Central services for ABAP 48
- Central services for Java 48
- Central Services Instances 14
- centralized console 24
- certified storage subsystems 25
- Chipkill 22
- client connectivity 16
- cluster 5
- cluster configurations 52
- Cluster Connect dialog 159
- cluster heartbeat communication 29
- cluster management software 2, 15
- cluster node 72
- clustered Linux systems 16
- clustered system 17
- Clustering 21
- Clustering for scalability 21
- clustering for scalability 21
- Clustering technology 21
- cluster-internal communication 75
- command center 12
- commit point 47
- Common Internet File System 33
- commonly used hybrid solution 43
- communication (IP) recovery 17
- compatibility matrix 92–93
- continuous client connection 52
- core components 17
- Core Package Cluster 94
- core software stack 24
- core system components 17
- creating file systems 110
- critical applications 28
- critical business factor 7
- cron daemon 118
- current sizing 92

D

- D15 instance 83
- DAS technology 33
- data instance 45
- database 2
- Database administration 7
- database clusters 21
- database compatibility 26
- database consistency 50

- database gateway 46
- Database hierarchy 179
- database home directory structure 80
- Database Instance 14
- Database layer 12
- database manager 135
- Database processes 12
- Database replication 40
- Database resource 179
- database unavailability 87
- DB protection 17
- DB Reconnect feature 87
- DB2 Administration Server 132
- DB2 administration server 45
- DB2 binaries 80
- DB2 data partitioning feature 80
- DB2 Instance 177
- DB2 IPC 134
- DB2 LUW 129
- DB2 Setup wizard 130
- DB2 SQL replication 40
- dbm configuration 134
- default Root Tag 168
- designing considerations 2
- development environment 10
- Device Mapper Multipath 94
- Device Mapper multipath Recovery Kit 168
- Device-Mapper Multipath 67
- dialog instance 46
- Direct Attached Storage 32
- disaster recovery requirements 33
- disk controller 60
- disk mirror 25
- disk takeover time 40
- dispatcher 46, 48
- Distributed Replicated Block Device 38
- D-Link Gigabit Ethernet switches 62
- documentation 17
- Domain Name Server 172
- downtime 7
- driver chain 71
- driver layer 36
- dual power connections 22
- Dual-port host bus adapters 61
- DVEBMGS12 instance 83

E

- eliminate single point of failure 4

- elimination of single points of failure 14
- Enqueue replication server 83
- enqueue servers 46
- environment variables 243
- ERS28 instance 83
- ERS29 instance 83
- Ethernet network adapter 23
- existing 20
- existing infrastructure 20
- existing installation 91
- Extended measurement 4

F

- fail 21
- failed service 4
- failing adapter 31
- failover cluster 21
- fail-safe environment 21
- failure of a single adapter 31
- false failover 160
- false failovers 5
- fast data transfer 34
- fault resiliency 44
- fault tolerance 33
- fault tolerant hardware 19
- fault-detection mechanisms 5
- faulty state 230
- Fibre Channel adapters 99
- Fibre Channel switch 34, 64
- file system check 236
- file systems 17
- firmware upgrades 99
- Fix Packs 95
- Flashcopy 39
- flexible shared storage 72
- floor space 21

G

- generic resource 243
- Gigabit Ethernet switches 62
- Gigabit network 65
- Global mirroring 37
- GRUB 106
- Guest permission 241
- GUI administration 17
- GUI password file 242
- gzipped tar archive 254

H

- HA methodology 43
- HA SAP architecture 46
- hardware availability 4
- Hardware clustering 5
- Hardware maintenance 7
- hardware redundancy 48
- heartbeat communication 29
- high 21
- high availability architecture 2
- high availability cluster 21
- high availability concepts 2
- High Availability Disaster Recovery 40
- high availability goal 59
- high availability infrastructure 3
- high level definition 92
- high performance storage network 34
- higher overall bandwidth 31
- highly available architecture 19
- host based authentication 80
- host based replication 38
- host bus adapter 69
- host bus adapters 23

I

- I/O modules 23
- IBM RDAC driver 68
- IBM TotalStorage 68
- IBM X-Architecture 20
- implement clustering 4
- implementing redundancy 20
- increase network availability 28
- industry-standard 20
- Information Integration 10, 12
- Informix Data Server 41
- infrastructure components 45
- initiating automatic instance restart 44
- Insight Collector 91
- installation 109
- installation directory 80
- instance specific data 47
- integrated runtime 10
- intelligent processes 5
- Intelligent switchback 167
- interfaces 17
- internet graphic service 46
- Internet Small Computers System Interface 33
- IP Application Recovery Kit 17

IP Local Recovery feature 174

J

Java central services 83
Java Cryptography Extension 97
Java enqueue servers 81
Java instances 216
Java processes 87
Java replication server 202
Java SDK 97
Java stacks 143
Java usage type 85
Java Virtual Machine 159

K

Knowledge Management 12

L

level of redundancy 20
licensing utilities package 152
LifeKeeper Core 152
LifeKeeper DB2 recovery kit 51
LifeKeeper for Linux Core 16
LifeKeeper GUI 16, 156, 240
LifeKeeper GUI Server 240
LifeKeeper log 251
LifeKeeper resources 251
LifeKeeper RPM packages 152
LifeKeeper server daemon processes 155
Link monitoring 66
Linux boot loader 106
Linux manual page multipath(8) 67
Linux standard kernel 67
Linux, UNIX, and Windows 40
Local Area Network 23
local HA availability 41
lock entries 213
lock table 50
Locking 46
log file types 252
logged transactions 40
logical unit of work 47
logical volume management 107
Logical Volume Manager 70, 236
loss of data 32
LVM Recovery Kit 168

M

maintenance session 258
maintenance task 259
major number 228
managed cluster 5
masking single point of failure 4
mass test locking 205
Master Data Management 12
mdadmd daemon 69
Media Access Control address 30
media kit 137
Memory ProteXion 22
Memory scrubbing 22
message server 46
Metro mirroring 37
micro-code feature 41
minimal host impact 41
minimal interruption 51
minimizing single point of failures 4
minor number 228
mirroring 69
mission-critical system 15
Multi Tiers 13
multipathing 20
multiple adapter cards 60
multiple fabrics 23, 34
multiple heartbeats 5
Multiple host bus adapters 34
multiple hot-swappable blades 20
multiple occurrences 84
multiple RDBMS vendors 40
multiple servers accessibility 33
multiple switches 34
multiple TCP/IP addresses 31

N

NetWeaver binaries 80
network adapters 112
network and storage switching 21
Network Attached Storage 32
network attached storage 24
network based replication 38
network configuration 111
Network design 28
Network File System 14, 33
network interface 32
NFS Server Recovery 17
NFS Server resource 182

O

- on-board network adapters 60
- One Tier 13
- OpenSSH 119
- operability of scripts 190
- operating or trunking 60
- operating system 17
- operation and the administration 6
- operation aspects 223
- operation mode 60
- Operator permission 241
- optimized for sharing 34
- OSI layer-2 Spanning Tree Protocol 29
- outage on a production system 86

P

- partitioning 105
- PCI bridge 60
- People Integration 10, 12
- performance clusters 21
- Persistent change 233
- physical interface 224
- physical topology 59
- planned and unplanned outages 7
- plug-in cards 60
- port trunking 64
- Pre-Extend Wizard 177
- prerequisites 90, 94
- Presentation layer 12
- Presentation processes 12
- preventing failures 19
- Primary Slave 225
- problem determination 263
- Process Integration 10, 12
- public key authentication 80
- public keys 121

Q

- Qlogic Fibre Channel 62
- Qlogic host bus adapters 122
- quality of service 64
- quality of service routing 64
- quickCheck script 186, 243

R

- RAID 0 (striping) 35
- RAID 1 (mirroring) 35
- RAID 10 (or RAID 1+0) 36
- RAID 5 (striping with checksum) 35
- RAID functionality 28
- RAID-1 disk mirroring 22
- RDB lock table 205
- recorded errors 209
- recovering failed applications 5
- recovery functions 19
- Recovery script 244
- recovery script 186
- recovery software 17
- Redbooks Web site 278
 - Contact us xxiv
- reduce complexity 20
- reduced downtime 24
- redundant backup 4
- Redundant hardware 15
- redundant signals 5
- registry variable 135
- reliability and availability 19
- remote management 63
- replication mechanisms 33
- replication methodologies 41
- replication solution 40
- replication table 50
- Replication technologies 40
- re-route or reconnect method 51
- restart the application 17
- rolling fixpack upgrades 40
- root cause analysis 7
- RPM database 254

S

- SAN volume controllers 36
- SAP administration user 145
- SAP application data 45
- SAP Application Platform 11
- SAP application suite 1
- SAP business solution 45
- SAP Central File System 48
- SAP certified maintenance levels 45
- SAP client 200
- SAP J2EE engine 11
- SAP Kernel level 92
- SAP NetWeaver architecture 21

- SAP NetWeaver implementation 89
- SAP Solutions 10
- SAP system administration 201
- SAP Web dispatcher 46
- SAPinst 139
- Scalability 15
- scalability 20
- scalable system 4
- SCS11 instance 83
- SCSI disk subsystem 17
- SCSI I/O driver 61
- SCSI locking 25, 75
- SCSI2 reservations 25
- SCSI3 persistent group reservations 25
- Secure Shell Daemon 120
- security requirements 52
- Server Fibre Channel host adapter 34
- server hardware 59
- server nodes 5
- shared directories 47
- shared storage 2
- shared-nothing architecture 44
- simplify solutions 21
- single point of access 12
- single points of failures 4
- sizing process 90
- Slackware 15
- software binaries 80
- Software clustering 5
- software deployment 46
- software deployment manager 49
- software infrastructure 17
- Software maintenance 7
- Software RAID 28
- Software RAID Recovery Kit 168
- Software upgrades 7
- solution management tools 21
- spool or update 46
- stand-by interface 66
- standby technologies 41
- static addresses 113
- stop and restart 17
- stop the application 17
- Storage Area Network 23, 32, 99
- storage integration 99
- storage subsystem 37
- storage volume 36
- store application data 34
- sub-system device driver 68

- SUSE Linux Enterprise Server 15, 68
- switchover cluster 14, 21
- synchronous data 37
- System availability 4
- System Central Services 46
- system clock 104
- system failure 5
- system uptime 16
- system wide availability 19
- System x servers 20

T

- technology solution 2
- template scripts 186
- test environment setup 81
- Test failover scenarios 197
- Three Tiers 13
- threshold of errors 22
- tips and techniques 263
- transaction locks 50
- transparent switch 15
- troubleshooting 244
- trunking mode 60
- TTY comm path 160
- TTY communication path 75
- Two Tiers 13

U

- uninterruptable power supplies 36
- uninterrupted data access 16
- unplanned outages 4, 203
- UUID 127

V

- validation process 92
- Vendor-specific drivers 69
- virtual block devices 68
- virtual interface 224
- virtual network console 100
- Virtualization 15
- VNCserver 241
- volume group 107

W

- Web server 201
- Websphere Q replication 40
- Weighted load-balancing 66

work processes 46, 48
workload characteristics 20

X

XFS file system 105

Y

YaST installer 102

Z

zero footprint client 80



Building High Availability with SteelEye LifeKeeper for SAP NetWeaver on SUSE Linux Enterprise Server

(0.5" spine)
0.475" <-> 0.875"
250 <-> 459 pages



Building High Availability with SteelEye LifeKeeper for SAP NetWeaver on SUSE Linux Enterprise Server

Architecting SAP NetWeaver for High Availability

Building a LifeKeeper cluster on Linux

Failover scenarios, administration, and troubleshooting

Business processes based on the SAP NetWeaver platform often require a high level of availability and fault tolerance.

Availability can be defined as the amount of time that application services are accessible to the end user, and is measured by the percentage of time that the application is available to the user. The application is highly available when it gets closer to the difficult-to-achieve 99.999% threshold of availability, also known as the five 9s of availability.

The Novell SUSE Linux Enterprise Server base operating system, SteelEye LifeKeeper for Linux cluster software, and SAP NetWeaver software provide capabilities that can be implemented to build a topology that fulfills these requirements.

In this IBM Redbooks publication, we discuss the concepts of high availability, provide an overview about the main components, and explain high availability topologies and implementation.

We cover these topics: Server hardware configuration, Novell SUSE Linux Enterprise Server software installation, SteelEye LifeKeeper cluster software installation, Network topology and configuration, Storage topology and configuration, DB2 software installation, and SAP NetWeaver software installation.

We also discuss the integration, test, and operation of all these components.

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