



SAP on DB2 UDB for OS/390 and z/OS: High Availability Solution Using System Automation

New enqueue instance implementation

System Automation
customizations for SAP high

Failover scenarios



Viviane Anavi-Chaput
Andre Borkenfeld
Florence Dubois
Don Geissler

Lars Morten Haga
Matthias Hein, Wolfgang Reichert
Udo Wandrer, Mike Zazulak



International Technical Support Organization

**SAP on DB2 UDB for OS/390 and z/OS: High
Availability Solution Using System Automation**

August 2002

Archived

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

First Edition (August 2002)

This edition applies to IBM z/OS V1R2 (5694-A01), IBM z/VM V4.2 (5739-A04), SuSE Linux Enterprise Server 7 for IBM zSeries (64-bit) , IBM System Automation for OS/390 V2.1(5645-006), IBM Tivoli Netview for OS/390 (5697-B82), IBM DB2 UDB for z/OS V7.1 (5675-DB2), and SAP R/3 4.6C SR2.

Note: This book is based on a pre-GA version of a product and may not apply when the product becomes generally available. We recommend that you consult the product documentation or follow-on versions of this redbook for more current information.

© Copyright International Business Machines Corporation 2002. 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

Notices	ix
Trademarks	x
Preface	xi
The team that wrote this redbook	xi
Become a published author	xiii
Comments welcome	xiv
Chapter 1. Introducing high availability and automation for SAP	1
1.1 High availability definitions	4
1.1.1 Degrees of availability	4
1.1.2 Types of outages	5
1.2 High availability and automation objectives for SAP	6
1.2.1 No planned outages	6
1.2.2 Transparent failover	7
1.2.3 Reduced operator errors	7
1.2.4 Health check for application problems	7
1.3 SAP high availability solution overview	8
1.3.1 SAP high availability	8
1.3.2 Automation of SAP high availability	9
1.3.3 Benefits of System Automation for OS/390	9
Chapter 2. Architecture of the SAP high availability solution.	11
2.1 Architecture components	12
2.1.1 New SAP enqueue instance replacing central instance concept ..	12
2.1.2 Network	16
2.1.3 File system	21
2.1.4 Database	23
2.1.5 Application design	24
2.2 Failure scenarios and impact	25
2.2.1 Old-style central instance without data sharing	26
2.2.2 Data sharing, sysplex failover, double network (single central instance)	28
2.2.3 Enqueue replication, NFS failover—full-blown HA	30
Chapter 3. Planning and preparing for the SAP high availability solution	33
3.1 SAP high availability solution configuration	34
3.1.1 Hardware requirements	34
3.1.2 Software requirements	36

3.2	Naming conventions	37
3.3	Network configuration	40
3.3.1	Network hardware components	40
3.3.2	Networking software components	41
3.4	File system setup	46
3.4.1	SAP directory definitions	48
3.5	DB2 setup	52
3.5.1	DB2 data sharing	52
3.5.2	ARM policy	52
3.6	Integrated Command Level Interface Server (ICLI Server)	53
3.7	Netview and System Automation basic setup	54
3.8	SAP installation	55
3.8.1	New SAP system installation	55
3.8.2	Preparing an existing SAP system for HA	61
3.8.3	SAP license	62
3.8.4	SAP logon groups	63
	Chapter 4. Customizing SAP for high availability	65
4.1	Installing and configuring the enqueue instance	66
4.1.1	Getting the standalone enqueue server code from SAP	66
4.1.2	Configuring the enqueue instance	66
4.1.3	SAP profile parameters	68
4.2	Configuring SAP for System Automation	71
4.2.1	ICLI servers	72
4.2.2	Enqueue instance	73
4.2.3	Application server instances	76
4.2.4	saposcol	78
4.2.5	rfcoscol	79
4.2.6	saprouter	81
4.2.7	Summary of start, stop and monitoring commands	82
	Chapter 5. Customizing System Automation for OS/390 for SAP	85
5.1	Preparing SA OS/390 for SAP HA	86
5.1.1	Before you start	86
5.1.2	Setting initialization defaults for SA OS/390 (AOFEXDEF)	86
5.1.3	Setting the region size for NetView to 2 GB	87
5.1.4	Customizing Status Display Facility (SDF)	87
5.1.5	Sending UNIX messages to the syslog	89
5.1.6	Setting MAXFILEPROC in BPXPRMxx	89
5.2	Defining the SAP-related resources	89
5.2.1	Overview of the resources	89
5.2.2	Naming convention	90
5.2.3	Classes	90

5.2.4 Database server	92
5.2.5 Enqueue instance and enqueue replication server	96
5.2.6 Application servers	107
5.2.7 SAP RED local applications	113
5.2.8 NFS server	116
5.2.9 saprouter	117
5.2.10 SAP local application	120
5.2.11 Defining superior groups	122
5.3 Overall picture	125
5.4 Summary tables	125
5.4.1 Classes	126
5.4.2 Applications	126
5.4.3 Application groups	127
5.5 Extension for DFS/SMB	128
5.5.1 Additions to the SA OS/390 policy	128
5.5.2 Additions to SDF	130
5.5.3 Additions to the Automation Table	130
Chapter 6. Verification procedures and failover scenarios	133
6.1 Overview of the test scenarios	134
6.1.1 Classification of the test scenarios	134
6.1.2 Test scenarios to verify the SA OS/390 policy	134
6.1.3 Scenarios that we tested	136
6.2 Test methodology	137
6.2.1 Purpose of the test	137
6.2.2 Expected behavior	137
6.2.3 Setup of the test environment	137
6.2.4 Verification of resource status	137
6.2.5 Preparation for the test (unplanned outage only)	142
6.2.6 Execution of the test	145
6.2.7 Verifications after the test	146
6.2.8 Analyze problems	147
6.3 Planned outage scenarios	147
6.3.1 Stop and start of the entire SAP RED system	147
6.3.2 Startup of all LPARs one after the other	152
6.3.3 Shutdown and restart of an LPAR	153
6.4 Unplanned outage scenarios	162
6.4.1 Failure of the enqueue server	163
6.4.2 Failure of the message server	170
6.4.3 Failure of the ICLI server	174
6.4.4 Failure of the NFS server	177
6.4.5 Failure of a TCP/IP stack	180
6.4.6 Failure of an LPAR	185

Chapter 7. Problem determination methodology	191
7.1 SA/390 problem determination	192
7.1.1 NetView netlog	192
7.1.2 z/OS SYSLOG	192
7.1.3 Message Processing Facility	193
7.1.4 Problem determination in System Automation for OS/390	193
7.1.5 UNIX messages	197
7.1.6 If nothing happens	198
7.1.7 When you are really lost	198
7.1.8 Get help from the Web	199
7.2 Where to check for application problems	199
7.3 Checking the network	201
7.4 Checking the status of the Shared HFS and of NFS	205
7.5 Checking the status of DB2 and SAP connections	205
Chapter 8. Change management	209
8.1 Updating the SAP kernel	210
8.1.1 Updating the SAP kernel (release 4.6)	210
8.1.2 Rolling kernel upgrade	212
8.2 Updating the ICL client and server	213
8.2.1 Rolling upgrade of the ICL client	213
8.2.2 Rolling upgrade of the ICL server	214
8.2.3 Updating to an ICL server with different protocol version	215
8.3 Updating DB2 or z/OS	215
Appendix A. Network setup	217
A.1 z/OS network settings	217
A.1.1 z/OS network settings shared by both LPARs	217
A.1.2 z/OS LPAR SC42	223
A.1.3 z/OS LPAR SC04	228
A.1.4 OSA-Express Fast Ethernet definitions for z/OS	232
A.2 Linux for zSeries network settings	233
A.2.1 Zebra setup - OSPF	233
A.2.2 Zebra setup - Zebra	233
A.3 Domain Name Server (DNS) definitions	234
Appendix B. File system setup	235
B.1 NFS server procedure	236
B.1.1 NFS export file	236
B.1.2 NFS attribute file	237
B.2 NFS client procedure	238
B.3 Mount commands on Linux /etc/fstab	238
B.4 BPXPRM member in SYS1.PARMLIB	239

Appendix C. DB2 and ARM setups	241
C.1 DB2 ZPARMS	242
C.1.1 ZPARM for subsystem D7X1	242
C.1.2 ZPARM for subsystem D7X2	248
C.2 The ARM policy	255
C.2.1 The ARM policy JCL	255
Appendix D. NetView and System Automation for OS/390 basic setup	257
D.1 Status Display Facility definition	258
D.1.1 AOFPSYST	258
D.1.2 AOF SAP	261
D.1.3 AOFTSC04	264
D.2 Sample REXX procedure	266
D.2.1 SANCHK	266
Appendix E. SAP customization for high availability	269
E.1 Shell scripts used for automation	269
E.2 SAP profiles	274
Appendix F. Additional material	279
Locating the Web material	279
Using the Web material	279
How to use the Web material	280
Related publications	281
IBM Redbooks	281
Other resources	281
SAP notes	281
Referenced Web sites	282
How to get IBM Redbooks	282
IBM Redbooks collections	282
Index	283

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

Trademarks

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

AIX®
CICS®
CUA®
DB2®
DFS™
DFSMS/MVS®
DRDA®
IBM®
IMS™
MVS™

NetView®
OpenEdition®
OS/390®
PAL®
Parallel Sysplex®
Perform™
RACF®
Redbooks(logo)™ 
RETAIN®
RMF™

SAA®
S/390®
SOM®
SP™
Tivoli®
VTAM®
z/OS™
z/VM™
zSeries™
3090™

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

Lotus®

Word Pro®

Notes®

The following terms are trademarks of other companies:

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

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

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

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

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

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

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

Preface

This IBM Redbook explains the SAP high availability with System Automation solution. This solution provides the means for fully automating the management of all SAP components and related products running on OS/390, z/OS, AIX, Windows, or Linux. The automation software monitors all resources and controls the restart and/or takeover of failing components, thereby ensuring near continuous availability of the SAP system.

The book describes the following:

- ▶ Requirements and objectives of high availability and automation for SAP
- ▶ Architecture of the SAP high availability solution
- ▶ Planning and preparation steps
- ▶ SAP customization for high availability
- ▶ System Automation customizations for SAP high availability
- ▶ Failover scenarios
- ▶ Problem determination
- ▶ Change management

The team that wrote this redbook

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

Viviane Anavi-Chaput is a Senior IT Specialist for BI, ERP and DB2 at the IBM International Technical Support Organization, Poughkeepsie Center. She writes extensively, teaches worldwide, and presents at international conferences on all areas of Business Intelligence and DB2 for OS/390. Before joining the ITSO in 1999, Viviane was a Senior Data Management Consultant at IBM Europe, France. She was also an ITSO Specialist for DB2 at the San Jose Center from 1990 to 1994.

Andre Borkenfeld is an IT Architect working for IBM Global Services in Germany since 1996. He is a certified SAP R/3 Basis Consultant. His areas of expertise include AIX, Linux on zSeries, Oracle, and DB2 for OS/390 and z/OS. Andre studied business information management at the University of Münster.

Florence Dubois is an IT Specialist for DB2 on z/OS at the EMEA ATS Products and Solutions Support Center, Montpellier, France. She has been with IBM since

1997, taking part in SAP performance benchmarks and providing onsite performance reviews to large SAP customers in EMEA. Her areas of expertise include data sharing and performance and tuning.

Don Geissler is an IBM Consulting Field Technical Sales Specialist in ebusiness Technical Sales for the Americas. He has been with IBM since 1983, providing systems engineering support to large S/390 customers. In September 1996 Don became a member of the IBM/SAP Competency Center, working with SAP R/3 on DB2 for OS/390 and z/OS. His primary responsibility is to consult with SAP R/3 on z/OS customers in the areas of high availability, data sharing, database backup and recovery, and disaster recovery.

Lars Morten Haga is a Systems Programmer at BBS A/S in Oslo, Norway. He has eight years of experience in S/390 systems. Recently he has been deeply involved in implementing SAP at his company, and he also has Linux experience.

Matthias Hein is an IT Specialist working for IBM zSeries field technical sales support in Germany. He has seven years of experience with Linux and three years of experience with S/390 and zSeries. He holds a PhD in physics from Heinrich Heine Universität, Düsseldorf. His areas of expertise include SAP on z/OS and on Linux on zSeries.

Wolfgang Reichert is a Senior Software Engineer in the German IBM development laboratory. Since 1995 he has been a member of the joint IBM/SAP platform team. In his position as solution architect, he is responsible for enabling the SAP solution on IBM zSeries. He was responsible for porting the SAP application server to z/OS. He also has designed the HA solution described in this redbook.

Udo Wandrer is an IT Specialist at IBM Global Services Germany. He has more than 20 years of experience in the large systems area. Mostly working in projects at customer sides, his main area of expertise is in NetView and System Automation. He has contributed to several redbooks dealing with NetView and System Automation for OS/390.

Mike Zazulak is an IBM Technical Sales Specialist in ERP Solutions Sales of North America. He has been with IBM since 1982. Since September 1998, Mike has been a member of the ERP Competency Center S/390 and zSeries team, working with SAP R/3 for DB2 UDB for OS/390 and z/OS. His primary responsibilities are supporting the sales team and customers with SAP R/3 for DB2 UDB on OS/390. Prior to this he was a member of the joint IBM/SAP DB2 UDB Porting Team in Walldorf, Germany where the SAP R/3 was ported to run on DB2 UDB for OS/390 and z/OS.

Special thanks to the following people for their support in setting up the system environment:

David Bennin, Richard Conway, Roy Costa, Robert Haimowitz, William G. White
IBM International Technical Support Organization, Poughkeepsie, NY, USA

Roland Haibl
IBM Germany

Ronald Northrup
IBM zSeries Parallel Sysplex PDT, Poughkeepsie, NY, USA

Mark Perry
Consultant, Germany

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

Guenter Zachmann, Jens Brauckhoff, Carl Philipp Staszkiwicz
SAP development team, Germany

Michael Gordon, Patrick Horkan
IBM USA

Thomas Gnech, Namik Hrle, Gerold Kurth, Raimund Thielker
IBM Germany

Tim Bohlsen
Consultant, Australia

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

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll 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 Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

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

ibm.com/redbooks

- ▶ Send your comments in an Internet note to:

redbook@us.ibm.com

- ▶ Mail your comments to:

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

Introducing high availability and automation for SAP

Since the initial announcement of SAP on DB2 UDB for OS/390 and z/OS¹ we have used DB2 Parallel Sysplex data sharing combined with the SAP sysplex failover feature to remove the database server as a single point of failure. This also gave customers the ability to avoid planned and unplanned outages of the database server.

The SAP high availability solution presented in this redbook further enhances this capability by removing the SAP Central Instance as a single point of failure and providing a means to automate the management of the enqueue server, message server, gateway server, and other SAP components for planned and unplanned outages.

The SAP high availability solution provides the means for fully automating the management of all SAP components and related products running on z/OS, AIX, Windows, or Linux. The automation software monitors all resources and controls the restart and/or takeover of failing components, thereby ensuring near continuous availability of the SAP system.

The availability of the enqueue server is extremely critical for an SAP system. If it fails, most SAP transactions will also fail. To address this single point of failure, SAP, in close cooperation with IBM, has changed the architecture of the enqueue

¹ Whenever the term “z/OS” is used it applies to both OS/390 and z/OS unless otherwise noted.

server. It is no longer part of the so-called "central instance". That is, it no longer runs inside a work process, but can be viewed as a separate enqueue instance. The enqueue server transmits its replication data to an enqueue replication server, which normally resides on a different system. The enqueue replication server stores the replication data in a shadow enqueue table that resides in shared memory. For a more detailed description of the new enqueue server and replication server, see 2.1.1, "New SAP enqueue instance replacing central instance concept" on page 12.

If the enqueue server fails, it is quickly restarted by System Automation for OS/390 and uses the replicated data in the shadow enqueue table to rebuild the tables and data structures. This means that a failure of the enqueue server is transparent to the end user and the SAP application. For a more detailed description of this process, see Chapter 2, "Architecture of the SAP high availability solution" on page 11.

The new architecture of the enqueue server is the key element of the high availability solution presented in this redbook. The description is built around a sample configuration that can be seen as a proposal and case study for the implementation of a SAP system on DB2 UDB for OS/390 and z/OS that provides for near continuous availability.

The solution is applicable to a homogeneous z/OS environment as well as to a heterogeneous environment. However, the described solution implies that all critical components are running on z/OS. The automation and availability aspects apply primarily to the components on z/OS, although the components on other systems are managed remotely.

As automation software, the IBM product System Automation for OS/390 was chosen because it not only provides the means for the implementation of a high availability system, but also includes all the features needed to streamline daily operations, for example features for automated startup, shutdown, and monitoring of the components of an SAP system and its dependent products.

The concept of dynamic Virtual IP Addresses (VIPA), together with dynamic routing, is used for some components. A dynamic VIPA moves with the corresponding server application. The client does not need to know the physical location of the server; it knows the server just by the virtual address. With this approach, a failover of the new enqueue instance becomes transparent to the client application.

In the following sections we discuss:

- ▶ Degrees of high availability
- ▶ System automation definitions
- ▶ Objectives of the SAP high availability solution
- ▶ An overview of the SAP high availability solution

Archived

1.1 High availability definitions

In this section we define the terms used to indicate various degrees of availability. We also discuss two types of outages that affect availability, which customers must be aware of.

1.1.1 Degrees of availability

The terms *high availability*, *continuous operation*, and *continuous availability* are generally used to express how available a system is. The following is a definition and discussion of each of these terms.

High availability

High availability refers to the ability to avoid unplanned outages by eliminating single points of failure. This is a measure of the reliability of the 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. This can be accomplished by quickly restarting failed components using a tool such as System Automation for OS/390.

Continuous operation

Continuous operation refers to the ability to avoid planned outages. For continuous operation there must be ways to perform administrative work, and 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. Using DB2 data sharing with sysplex failover is an example of how this is accomplished in an SAP environment. The redbook *SAP R/3 on DB2 UDB for OS/390: Database Availability Considerations*, SG24-5690, describes how many planned outages can be avoided by taking advantage of DB2 data sharing and SAP sysplex failover.

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.

1.1.2 Types of outages

Because the availability of the SAP system is a critical business factor, and therefore the highest level of availability must be provided. Customers must be aware of the types of outages and how to avoid them. In this section we discuss planned and unplanned outages.

Planned outage

Planned outages are deliberate and are scheduled at a convenient time. These involve such activities 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 system component. They include hardware failures, software issues, or people and process issues.

In a report issued by Gartner Research, *Enterprise Guide to Gartner's High-Availability System Model for SAP*, R-13-8504 dated 12/20/01, they discuss the causes of application downtime (see Figure 1-1 on page 6). According to Gartner, one-fifth of unplanned outages result from hardware failure, network components, operating system problems, or environmental problems. In the case of hardware or software failures, the reliability and resilience of these components determines the impact of unplanned outages on the SAP system.

Two-fifths of unplanned outages result from application errors. These include software bugs, application changes, or performance issues.

The remaining two-fifths of unplanned outages result from operator errors and unexpected user behavior. These include changes to system components, not executing tasks or executing tasks improperly or out of sequence. In these cases the original outage could have been planned but the result is that the system is down longer than planned.

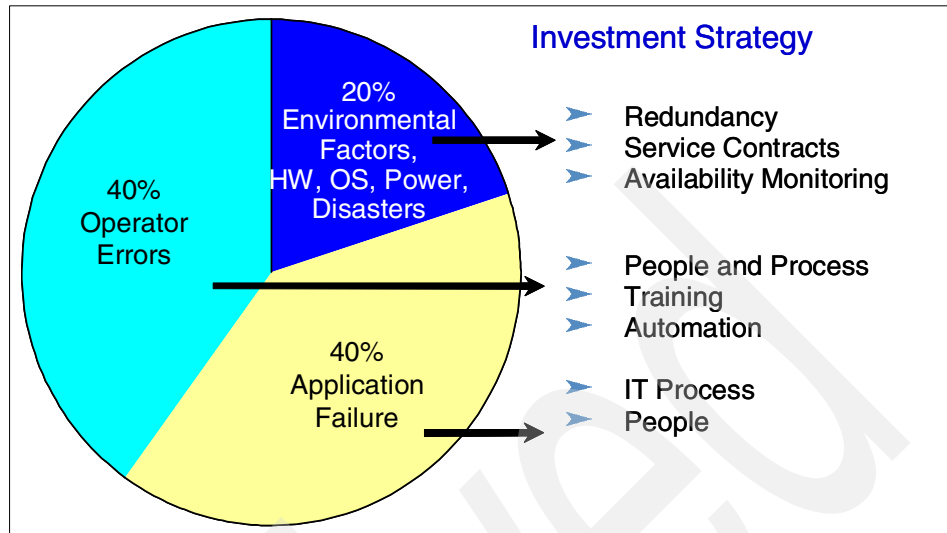


Figure 1-1 Causes of application downtime and appropriate responses

1.2 High availability and automation objectives for SAP

The objectives of the SAP high availability solution are to address the common causes of planned and unplanned outages by:

- ▶ Eliminating planned outages and providing continuous availability of the SAP system to end users
- ▶ Minimizing the effects of unplanned outages
- ▶ Reducing operator errors
- ▶ Monitoring the status of SAP application components

1.2.1 No planned outages

Planned outages for software or hardware maintenance can be avoided by using Parallel Sysplex data sharing and SAP sysplex failover to dynamically move remote application server instances to standby database servers. The procedures for doing this are documented in *SAP R/3 on DB2 UDB for OS/390: Database Availability Considerations*, SG24-5690.

SAP release 4.6 has added the capability to switch an application server instance between the primary and secondary database server using transactions DB2 or ST04. Prior to this the only way to switch database servers was to stop the active ICL. See SAP note 509529 for further details.

SAP release 6.10 further extends sysplex failover by adding the capability to define connections to multiple database servers. The application server instance cycles through the defined database servers in the event of a failure, or when using transaction DB2 or ST04. See SAP note 402078 for further details.

Planned outages for database administration can be avoided by utilizing DB2 online utilities such as image copy or reorg.

If the enqueue instance is on the system where maintenance is to be performed, system automation can be used to move the enqueue instance to a standby z/OS LPAR. This move is transparent to the end users. SAP work processes will automatically reconnect to the moved enqueue instance without failing any transactions.

1.2.2 Transparent failover

We have always had transparent failover for remote application server instances using Parallel Sysplex data sharing and SAP sysplex failover. Because of the newly designed enqueue server, the enqueue instance can now be moved or restarted transparent to the end users. SAP work processes automatically reconnect to the enqueue instance without failing any transactions.

1.2.3 Reduced operator errors

The SAP high availability solution uses System Automation for OS/390 (SA/390) to automate the starting, stopping, and monitoring of all SAP components. By automating daily operations, there is less opportunity for error when starting or stopping SAP components. SA/390 provides the ability to define component dependencies with parent-child relationships. In doing this, SA/390 checks that a component that has a parent is not started before its parent is active. SA/390 also checks that a component is not stopped if there are child components still active. This ensures that an orderly start or stop of the SAP system is accomplished with little opportunity for operator error. See Chapter 5, “Customizing System Automation for OS/390 for SAP” on page 85 for a description of how this is set up.

1.2.4 Health check for application problems

SAP now provides a utility, rfcping, to monitor the status of application servers. The SAP high availability solution uses System Automation for OS/390 (SA/390) to invoke the monitoring task at regular intervals to check the status of application server instances. The monitoring task issues an RFC call to the application server and waits for a response. If a response is received, then the monitor ends. If a response is not received, the monitor signals SA/390 that the

application server instance is down. For a more detailed description of rfcping, see “rfcping” on page 77.

1.3 SAP high availability solution overview

With the old-style central instance running on z/OS, a database connection was needed in order to start the central instance. The enqueue server ran in a work process under the control of the dispatcher. If the central instance fails, most transactions will fail. This causes a disruption of the whole SAP system.

The new enqueue server architecture enables the enqueue server to run as a standalone component. No database connection is required. The application server instances connect directly to the enqueue server. The message server is no longer in the communication path. The new architecture also adds an enqueue replication server that maintains a shadow enqueue table using data sent from the enqueue server.

If the enqueue server fails, the SAP work processes wait and reconnect to the enqueue server after it is restarted. When the enqueue server is restarted, it uses the shadow enqueue table to rebuild the enqueue data table. Transactions will not fail but will wait for the reconnection to the enqueue server. System Automation for OS/390 is used to automatically restart the enqueue server.

Because of this new architecture, we can now move the enqueue server from one z/OS LPAR to another in the same way we use SAP sysplex failover to move connections to the database server in a data sharing configuration.

We have combined the enqueue server, message server, gateway server, and syslog collector into a so-called enqueue instance. We use System Automation for OS/390 (SA/390) to automate the starting, stopping, and monitoring of the enqueue instance.

1.3.1 SAP high availability

As described in “High availability” on page 4, elimination of single points of failure is required. We use DB2 data sharing to remove the database server as a single point of failure. Now, with the new enqueue instance, the enqueue server has been removed as a single point of failure. The SAP high availability solution also adds a movable NFS server and dynamic virtual IP addressing for moving application components. System Automation for OS/390 is used to monitor these components and quickly restart them if they should fail.

1.3.2 Automation of SAP high availability

The SAP high availability solution uses System Automation for OS/390 (SA/390) to automate all SAP components. These include DB2 subsystems, ICLL servers, local and remote application server instances, enqueue server, message server, syslog collector and sender, gateway server, enqueue replication server, TCP/IP, and NFS server. By automating all the SAP components, the SAP system can be started, stopped, and monitored as a single resource. This provides for the highest level of availability by reducing operator commands, thus reducing the chance for operator errors.

1.3.3 Benefits of System Automation for OS/390

An SAP system has many components, and operation of these components is complex. There is a real need to simplify the operation of the SAP system. As more SAP systems are added, this need becomes even greater. Simplifying the operation of the SAP system can help you meet your service level agreements. It can also help you contain costs while more efficiently using your operations staff by removing repetitive tasks that are error prone.

System Automation for OS/390 (SA/390) offers system-wide benefits by simplifying the operation of the entire SAP system. This is particularly important when there are multiple SAP systems to manage. It is necessary for the various components of the SAP system to be started and stopped in the proper order. Failure to do this delays the system's availability.

In SA/390 the emphasis has switched from purely command-driven automation to goal-driven automation. Automation programmers now define the default behavior of the systems and application components in terms of dependencies, triggering conditions, and scheduled requests.

The impact of an unplanned incident is further mitigated by the speed of restarting and the degree of automation. The goal-driven design of SA/390 provides both the speed and a high degree of automation while avoiding the complexity of scripted automation tools, hence reducing automation errors.

The automation manager works to keep systems in line with these goals and prioritizes operator requests by using its awareness of status, dependencies, and location of all resources to decide what resources need to be made available or unavailable, when, and where. The number of checks and decisions it has to make can be very high. A human simply can't do the same as fast and reliably as the automation manager.

Goal-driven automation greatly simplifies operations. Operators just request what they want, and automation takes care of any dependencies and resolution of affected or even conflicting goals. Sysplex-wide automation can also remove the

need for specifying extra configurations for backup purposes. Instead, cross-system dependencies and server and system goals can be used to decide which backup system is to be chosen.

Given that the SAP system is generally critical to the operation of the business and that human errors can occur, the use of an automation tool that responds in a consistent way to a particular event can help deliver on the promise of continuous operation.

More information on System Automation for OS/390 can be found on the Web at:

<http://www.ibm.com/servers/eserver/zseries/software/sa>

Architecture of the SAP high availability solution

This chapter explains the architecture of the SAP high availability solution and its system infrastructure requirements.

We discuss the following:

- ▶ Architecture components
- ▶ Failure scenarios and impact

2.1 Architecture components

The SAP high availability solution involves the following architecture components:

- ▶ New SAP enqueue instance
- ▶ Fault tolerant network
- ▶ File system considerations
- ▶ Database considerations
- ▶ Designing applications for a highly available environment

2.1.1 New SAP enqueue instance replacing central instance concept

In the old design, the central instance provides the following functionality:

- ▶ It hosts the enqueue work process.
- ▶ It usually serves as location of the message server and the syslog collector.
- ▶ It hosts a gateway process and serves as primary destination for RFC connections.

Usually the SAP file systems physically reside on the same system where the central instance is running. The file systems are made available to other application servers by means of NFS.

For the high availability solution, the central instance has been disassembled and redesigned into standalone components. The independence of the components allows for more efficient recovery should a component become unavailable, and provides better performance of the enqueue services.

For the sake of simplicity, the following standalone components have been grouped together and make up the so-called *enqueue instance*:

- ▶ Enqueue server
- ▶ Message server
- ▶ Gateway (optional)
- ▶ Syslog collector (optional)

As members of an instance, the components share an instance directory and an instance profile. Nevertheless, the components can be started, stopped and recovered independently. None of them require access to the database.

Furthermore, the components of the enqueue instance share one virtual IP address (VIPA). With this approach the setup of TCP/IP and the SAP profiles is kept as small as needed. All the components benefit from an IP takeover simultaneously and in the same manner.

The message server, the gateway, and the syslog collector have been standalone components before. However, the enqueue server and its client/server protocol have been redesigned.

Old style enqueue services with the central instance

For comparison, the old architecture and request flow are described first.

As shown in Figure 2-1, the enqueue server resides inside a work process. The message flow goes from the requesting work process to its dispatcher, via the message server and the dispatcher of the central instance to the enqueue work process. The response message is sent back the same way.

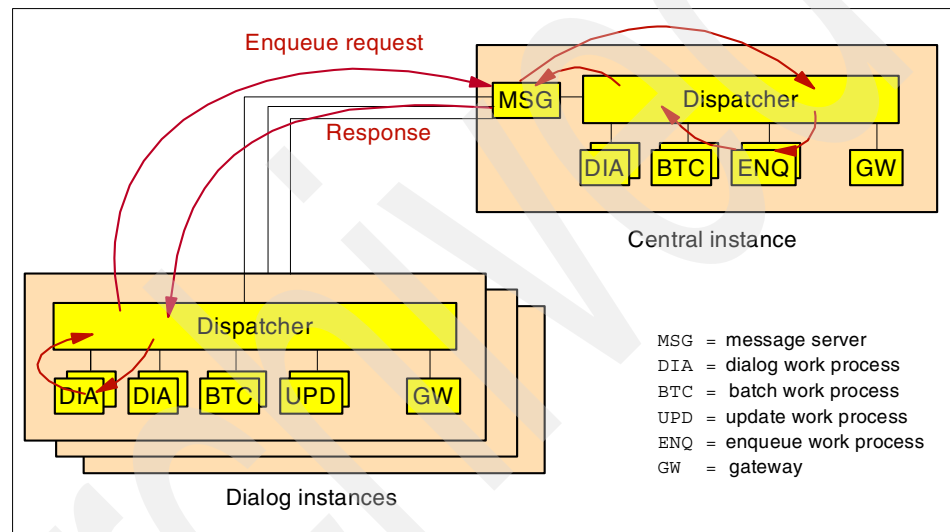


Figure 2-1 SAP enqueue services with the old central instance concept

Failure of any of the involved components (central instance, message server, enqueue work process) causes a disruption of the whole SAP system. For the recovery of the central instance, a working database connection is needed. Throughput is limited by the capacity of the message server and the dispatcher of the central instance.

New standalone enqueue server

The availability of the enqueue server is extremely critical for an SAP system; if the enqueue server cannot be reached, the SAP system is basically not operational, since most transactions fail to run.

The enqueue server has been redesigned to become a standalone component. It is no longer part of the central instance, that is, it no longer runs inside a work process. The new enqueue server does not require access to the database.

An application server instance connects directly to the enqueue server by using a virtual IP address (VIPA). The message server is no longer in the communication path. See Figure 2-2 on page 15.

To allow continuous availability and transparent failover, the *enqueue replication server* has been introduced. It is a standalone component as well. It connects to the enqueue server. When connected, the enqueue server transmits replication data to the replication server. The replication server stores it in a shadow enqueue table, which resides in shared memory. In case of a failure of the enqueue server, it is used to rebuild the tables and data structures for the enqueue server so it can be restarted.

If the enqueue replication server is unavailable, the SAP system continues to be up and running. However, there is no longer a backup for the enqueue server.

The enqueue replication server is not considered a member of the enqueue instance because it runs on a different system, though it may share the same instance directory and instance profile, providing that a shared file system is used.

The multi-threaded architecture of the standalone enqueue servers allows parallel processing and replication. The I/O processing for the TCP/IP communication, which caused the throughput limitations in the old design, is now distributed over several I/O threads. This, together with the elimination of the message server in the enqueue communication path, makes possible a significantly higher throughput.

Failover and recovery of the enqueue instance

Figure 2-2 on page 15 shows the principal TCP/IP communication paths between the application server instances and the enqueue and message servers. The other SAP components of the enqueue instance (gateway, syslog collector and sender) are not shown because they are of minor relevance for the failover scenario.

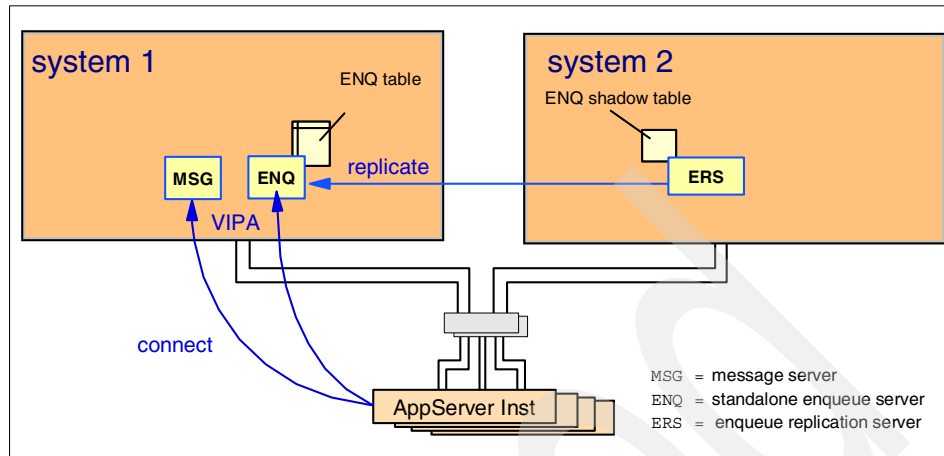


Figure 2-2 Initial startup of the enqueue instance

If the first system fails, the second system takes over the role of the first one, as shown in Figure 2-3:

1. The IP address (VIPA) is taken over.
2. Enqueue and message servers are restarted.
3. The enqueue table is rebuilt from the shadow table.
4. The application servers reconnect to the enqueue server and the message server.

The failover is fully transparent to the application. The enqueue locks are preserved and transactions continue to run.

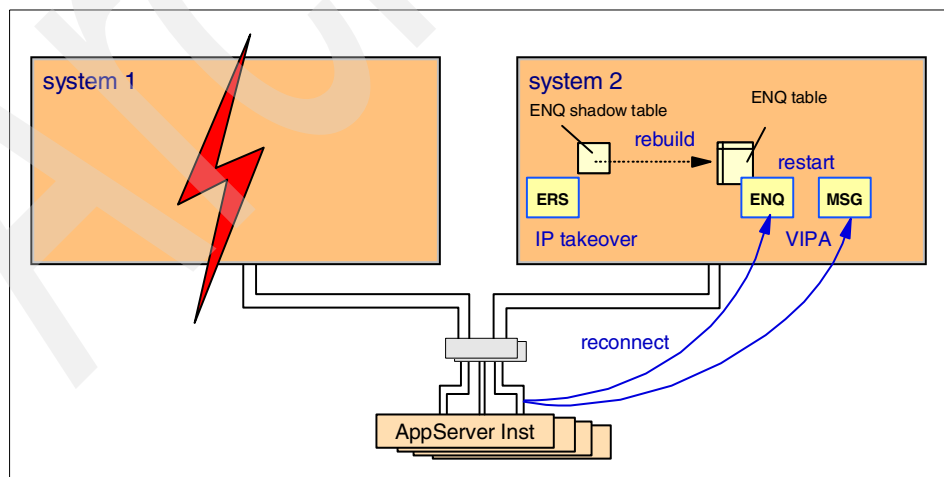


Figure 2-3 Failover of the enqueue instance and recovery of the enqueue table

After a successful failover of the enqueue server, the replication server is no longer needed on system 2 and therefore can be stopped. If another system is available or becomes available, the replication server is started on that system and a new shadow enqueue table is established. This is shown in Figure 2-4.

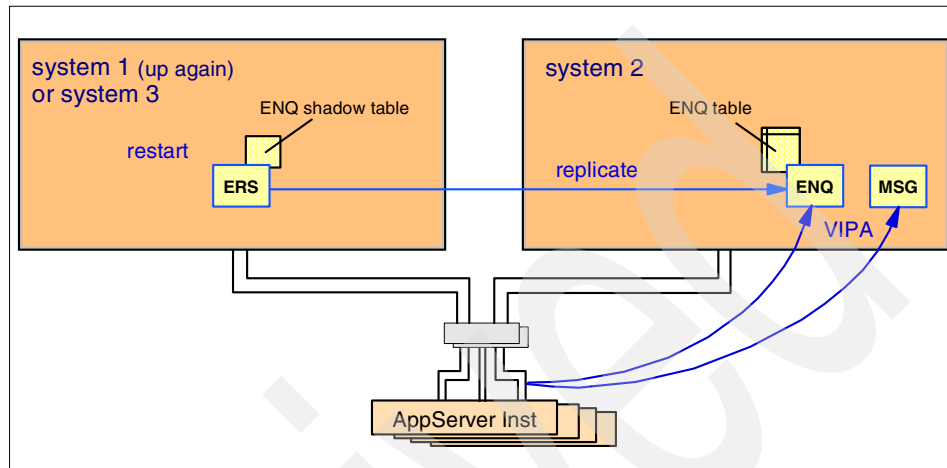


Figure 2-4 Movement of the enqueue replication server

2.1.2 Network

To protect against network failures, all network components need to be duplicated. IBM platforms (z/OS, Linux on zSeries, and AIX) support an elegant method for identifying the location of hosts and applications in a network: It is done by means of virtual IP addresses (VIPA).

Static VIPAs are used to locate a host while *dynamic VIPAs* can be activated by and moved with an application.

For a fault-tolerant network it is furthermore recommended to define a static VIPA together with the SOURCEVIPAs option for every participating system. The OSPF (Open Shortest Path First) routing protocol ensures that failures of any network component (network adapter cards, routers or switches, cables) are detected instantaneously and an alternative route is selected. This automatic rerouting is accomplished by the TCP/IP layer and is transparent to the application. TCP/IP connections are not disrupted.

Figure 2-5 on page 17 shows the general concept of a fault-tolerant network with duplicated network components and VIPA.

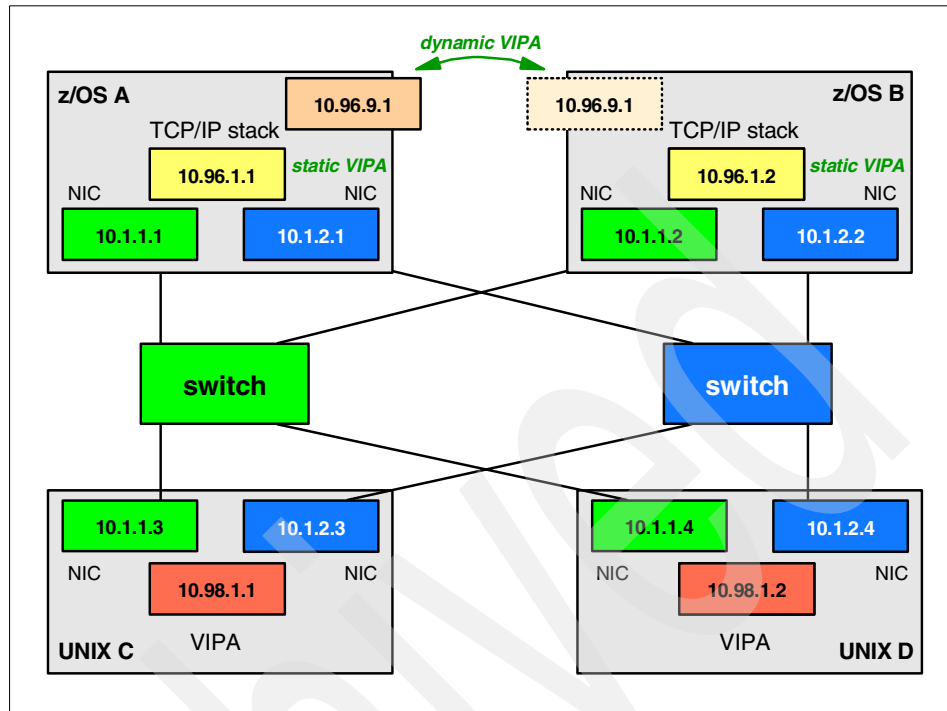


Figure 2-5 General concept of a fault-tolerant network

The following figures show how dynamic rerouting works. In Figure 2-6 on page 18 the virtual IP address `virt_addr_1` on system A can be reached through IP addresses `addr_1`, `addr_2` and `addr_3`. These real addresses are seen as gateways to the virtual IP address. ENQ and MSG indicate two applications running on that system. You can imagine that these are the SAP enqueue server and the message server.

Connections coming from application server instances choose `addr_1` or `addr_2` as gateway to system A. The third possible connection through system B is not chosen because OSPF selects the shortest path first.

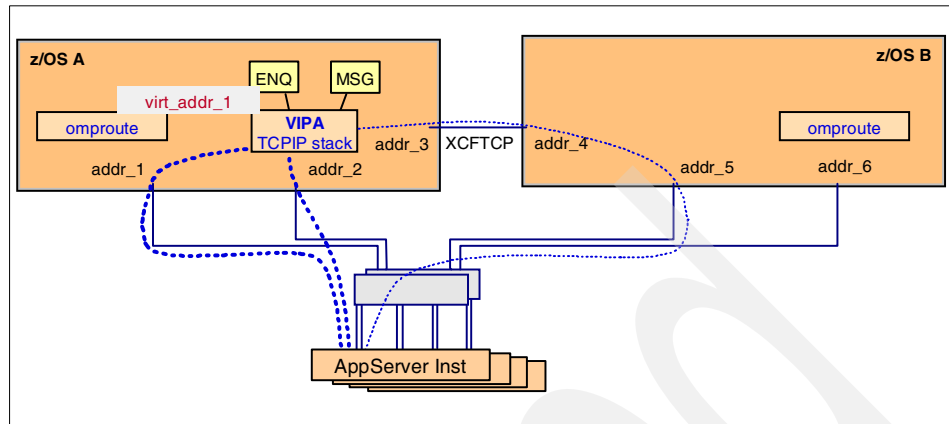


Figure 2-6 Alternative paths in a duplicated network

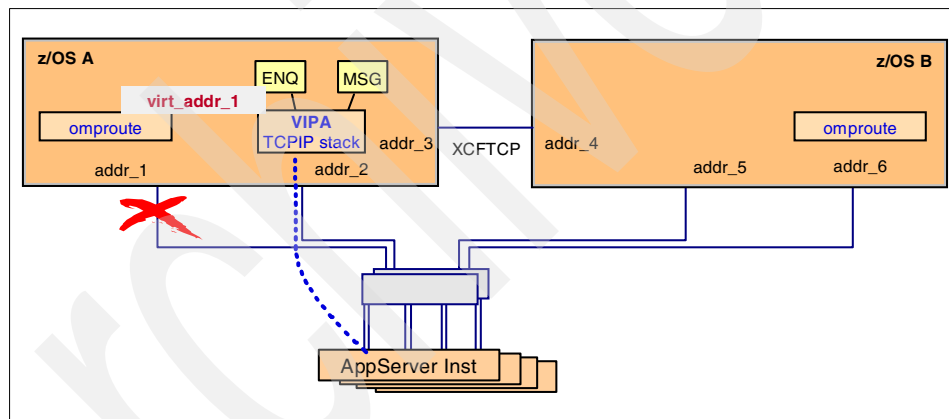


Figure 2-7 Rerouting if a network adapter card fails

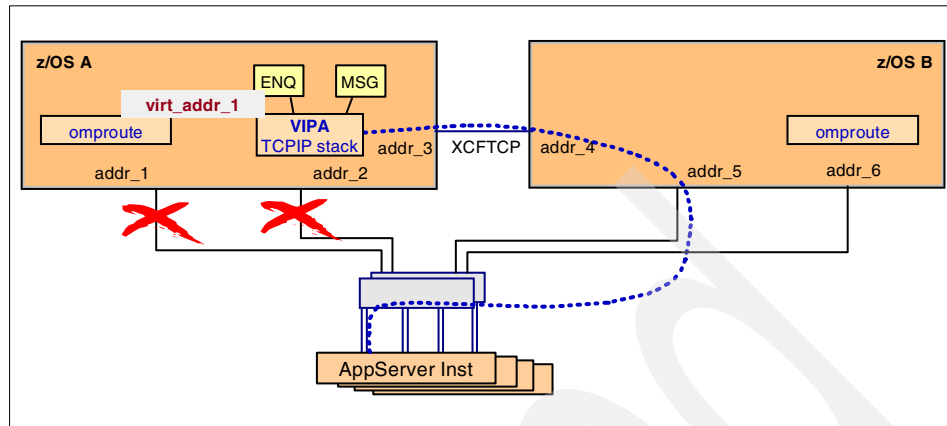


Figure 2-8 Rerouting in a Sysplex even in case of two failing network cards

What happens if network adapter card addr_1 fails? As shown in Figure 2-7 on page 18 there is still a path from application server instances to system A. All TCP/IP traffic is now routed through addr_2. The rerouting is absolutely transparent to the application. The router daemons on each system detect the missing links and propagate alternative routes. On z/OS, the router daemon is omproute.

What happens if network adapter card addr_2 fails, too? As shown in Figure 2-8, even then a path from application server instances to system A remains available. All TCP/IP traffic is now routed through system B via addr_3. Again, the rerouting is transparent to the applications.

Figure 2-8 also shows that, as long as any system in the sysplex is reachable, all systems are reachable. However, what happens in case of a TCP/IP or LPAR failure? The automation software is able to detect such a failure, move virt_addr_1 to system B, and restart the applications there. The takeover of the ENQ and MSG server together with the virtual IP address is shown in Figure 2-9 on page 20. Now addr_4, addr_5 and addr_6 are propagated as gateways to virt_addr_1. The IP takeover to another system disrupts existing connections. Application server instances have to reconnect and resynchronize their communication.

In a sysplex it can be ensured that the VIPA is really moved, that is, that it is certain to be deleted on system A, and that any connections to applications on system A using this VIPA are disrupted.

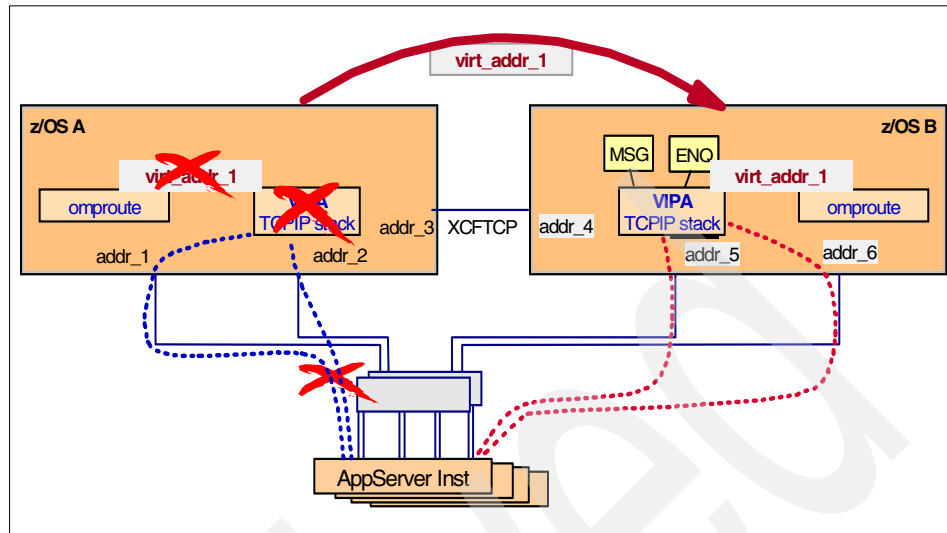


Figure 2-9 VIPA takeover and dynamic rerouting

In the scenario described in this redbook, the connections between Linux (hosting an application server) and z/OS (hosting the primary database server for this application server) take advantage of HiperSockets. The connection through the HiperSockets does not need any physical network adapter cards, routers, switches, or cables and therefore is an absolutely reliable connection. In this configuration, a VIPA definition on the Linux system is not needed with respect to the database connection, though it could be useful for incoming connections from the LAN.

Static VIPAs are used to connect to SAP components that are not moved between systems, like the ICL servers or the application server instances.

Dynamic VIPAs need to be defined for movable components, namely a dynamic VIPA is defined for each of the following resources:

- ▶ NFS server
- ▶ SAP enqueue instance
- ▶ SAP network interface router (saprouter)

While the rerouting shown in Figure 2-6 on page 18 through Figure 2-8 on page 19 is applicable to both static and dynamic VIPAs, the takeover shown in Figure 2-9 applies to dynamic VIPAs only.

2.1.3 File system

The SAP system requires shared access to some directories (global, profile, trans), while sharing is an option for other directories (for example, the directory containing the executables).

On z/OS, shared directory access is achieved with the Shared HFS feature.¹

In a heterogeneous environment, remote servers (such as Linux, AIX or Windows application servers) need access to the SAP directories as well.

In the case of UNIX systems, NFS is needed to share files. As a result, the availability of the file systems together with the NFS server becomes a critical factor. In this document it is assumed that the critical file systems reside on z/OS.

The z/OS file system can be made available as a network drive to Windows systems by using DFS SMB or SAMBA.

Important: File access is not transactional. There is no commit or rollback logic. In case of a system failure there is no guarantee that the last written data has been stored on disk. This is even more important for remote file access (NFS, FTP) where a disruption of the communication may result in an incomplete data transmission.

The methods described in this chapter ensure that the file systems become available again, quickly and automatically. In most cases this is transparent to the SAP system.

See also “Application design” on page 24.

Failover of the NFS server

NFS clients try to reconnect automatically if a connection is disrupted. When the NFS server fails, the NFS server can be restarted on the same system. If this is not possible, it is restarted on a second system.

To allow this failover to be transparent to applications on the NFS client side, the following conditions must be met:

¹ The name *Shared HFS* is a little bit confusing because it seems to imply that it is related to the HFS and only the HFS. However, the *Shared HFS* is a logical layer above the physical file system implementation. As physical file systems, all available file system implementations are supported, i. e. HFS, zFS, NFS (the client), TFS (the temporary file system), and DFS (the distributed file system). For the SAP directories HFS and zFS are appropriate.

- ▶ A dynamic VIPA is defined that moves with the NFS server.
- ▶ The physical file systems that are exported by the NFS server must also be accessible on the second system. This is another reason for using shared HFS.

The failover scenario is shown in Figure 2-10 and Figure 2-11 on page 23. Note that the NFS VIPA is different from the VIPA of the enqueue instance. So they can be handled independently of each other.

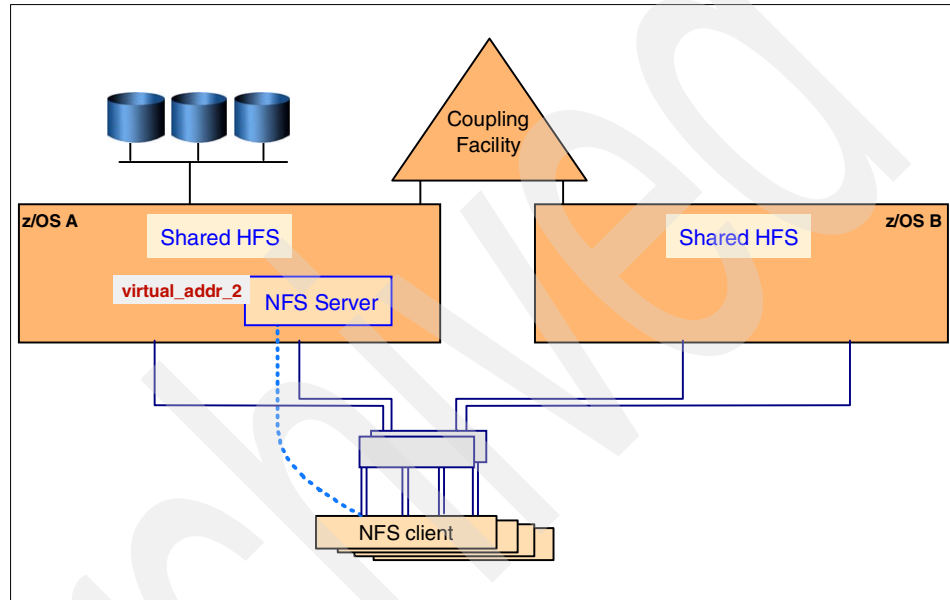


Figure 2-10 Initial NFS client/server configuration

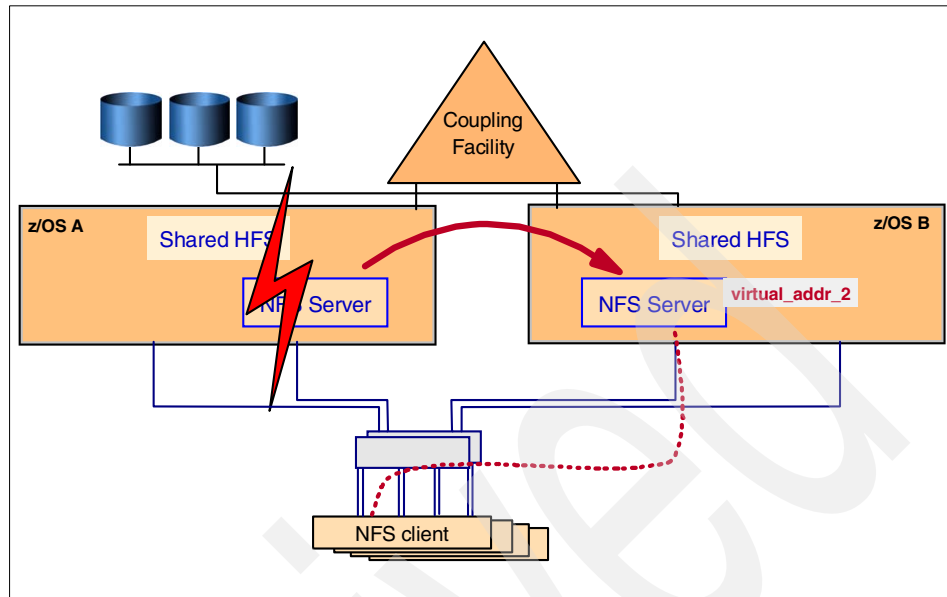


Figure 2-11 Failover of the NFS server

2.1.4 Database

The DB2 database server is one of the components of the SAP system that is critical to the availability of the SAP system. Other critical components are the enqueue server and the message server, which are discussed in “New SAP enqueue instance replacing central instance concept” on page 12.

If the database server is not available, the entire SAP system is unavailable. For this reason special attention should be paid to providing the ability to keep the database server available. Availability of the database server can be thought of in two degrees, high availability and continuous availability. High availability provides for the ability to reduce the impact of an unplanned outage such as a database server abend. Continuous availability provides for the ability to reduce the impact of both planned and unplanned outages.

For this redbook we used System Automation for OS/390 to provide the ability to automate the starting, stopping, monitoring, and restarting of the database server. With System Automation for OS/390 we are able to provide high availability for the non-data sharing configuration and continuous availability for the data sharing configuration.

The following sections discuss the impact of database server unavailability when running in non-data sharing and data sharing configurations.

Non-data sharing

In a non-data sharing configuration the database server is a single point of failure. Whenever it is unavailable, the entire SAP system is unavailable. There are two reasons why the database server might not be available: Planned and unplanned outages.

In this configuration the database server must be stopped whenever there is a need to upgrade or apply maintenance to it or the z/OS operating system. These are generally referred to as planned outages and are unavoidable but can be scheduled at a convenient time.

For unplanned outages of the database server there are several tools that can be utilized to minimize their impact. Several customers have been using OS/390 Automatic Restart Manager (ARM) for several years to quickly restart a failed DB2 system. There are also tools by other vendors that provide for quick restart of the database server.

System Automation for OS/390 provides the added advantage of automating daily operational activities such as starting, stopping, and monitoring the entire SAP system, including the database server. System Automation for OS/390 also ensures that components are started and stopped in the proper sequence. The automating of these activities provides for quicker SAP system startups with less errors, thus providing improved overall system availability.

Data sharing

A data sharing configuration eliminates the database server as a single point of failure and provides for near continuous availability. In a data sharing configuration, planned outages can be avoided by using the SAP sysplex failover feature to move workload off the DB2 member needing the outage to an available DB2 member in the data sharing group. In the case of an unplanned outage, the sysplex failover feature is used to switch the workload to a surviving DB2 member. In either situation, the SAP system remains available to the end users.

In a data sharing configuration, system automation becomes even more important because there are more database server components to deal with. As stated above, automating the daily operations of starting, stopping, and monitoring all the components of the SAP system provides for improved SAP system availability by eliminating most human errors.

2.1.5 Application design

The hardware, operating system, database, middleware, as well as the SAP components and applications, provide high availability features. Other

applications or connectors to be used in a high availability environment should also be designed with high availability in mind.

Therefore, when customers or their consultants design their own applications or write add-ons to existing applications, or buy them from other software vendors, it is good to consider the following recommendations:

- Make the applications restartable.

Consider that the application server instance or the system the application runs on may fail. Automatic restart of the same application on an alternative system can be accomplished with available job scheduling programs.

The data in the database is in a consistent state because any in-flight transactions get rolled back to the last commit point. So it is now the responsibility of the application to find out how far the work has proceeded and where to continue.

- Do not store vital data in files.

Instead, use the database. For transfer of data between applications, use the appropriate products, such as MQSeries, which provides transactional semantic and guaranteed delivery.

If you really think you need to transmit vital data from one application to another by use of files, then at least do the following:

- Check data completeness and integrity (for example, by calculating the checksum) before processing the data,
- Provide means to easily recreate the data in case errors are detected.

2.2 Failure scenarios and impact

This section discusses the impact of various failure scenarios on the SAP system end user. For all the configurations discussed we assume that System Automation for OS/390 (SA/39) is being used. Without SA/390 the impact on the SAP system would be much different from what is shown in the Impact column in the tables below. Without SA/390, all recovery actions would have to be done manually. Usually when things are done manually under the pressure of a system outage, recovery takes longer and is error prone. At best this would cause SAP transactions to timeout and roll back.

The scenarios discussed are those that are of most concern to customers. They are a subset of the scenarios discussed in Chapter 6, "Verification procedures and failover scenarios" on page 133.

In the following tables, SA indicates actions taken automatically and instantaneously by System Automation for OS/390, and Usr indicates actions

taken by the user. Also, for the action “Usr: Restart transactions” a customer could use workload scheduling software for this purpose (e.g., Tivoli Workload Scheduler). However, due to time constraints we were not able to test this.

The differences to the impact between the configurations are marked in *italic* font.

2.2.1 Old-style central instance without data sharing

In the scenario in Table 2-1, the SAP system is using the old style central instance and data sharing has not been implemented for the DB2 database server. Most customers are using this configuration today without system automation.

Summary: Database, central instance, and network are single points of failure. Failures of these critical components impact the whole SAP system.

Table 2-1 Simple configuration

Failure	Impact	Actions
DB2	<ul style="list-style-type: none"> ▶ Rollback of transactions ▶ Application servers wait until DB2 is up again 	SA: Restart DB2 Usr: Restart transactions
ICLI server	<ul style="list-style-type: none"> ▶ Rollback of transactions ▶ Application servers wait until ICLI server is up again 	SA: Restart ICLI server Usr: Restart transactions
Central instance	<ul style="list-style-type: none"> ▶ Rollback of transactions ▶ Application servers wait until central instance is up again 	SA: Restart central instance Usr: Restart transactions
Message server	<ul style="list-style-type: none"> ▶ Most transactions are inhibited because the enqueue work process is not reachable ▶ Application servers wait until message server is up again ▶ Group logon inhibited 	SA: Restart message server Usr: Restart transactions
Application server instance	<ul style="list-style-type: none"> ▶ Transactions on this instance are lost ▶ Rollback of database updates ▶ User sessions on this instance are lost 	Usr: connect to another instance Usr: Restart transactions SA: Restart instance

Failure	Impact	Actions
Gateway	<ul style="list-style-type: none"> ▶ For most transactions, no impact ▶ Connections to registered RFC servers inhibited until they have reconnected to gateway 	SA: Restart gateway
Syslog collector	<ul style="list-style-type: none"> ▶ For most transactions, no impact ▶ Global syslog file out of date 	SA: Restart syslog collector
saprouter	<ul style="list-style-type: none"> ▶ User sessions lost ▶ Reconnect inhibited 	SA: Restart saprouter Usr: Reconnect
NFS server	<ul style="list-style-type: none"> ▶ Some transactions stop, fail after timeout ▶ Batch transactions stop, fail after timeout ▶ Restart of application servers inhibited ▶ If data was written to file, last written data is in doubt 	SA: Restart NFS server Usr: Restart transactions
File system	<ul style="list-style-type: none"> ▶ Some transactions inhibited ▶ Batch transactions fail ▶ Restart of application servers inhibited ▶ If data was written to file, transaction is rolled back and last written data is in doubt 	Usr: Recover and remount the file system Usr: Restart transactions
Network (router, switch, adapter card)	<ul style="list-style-type: none"> ▶ Lost connectivity to message server and gateway server (see failures of these components) ▶ Rollback of transactions on remote application servers ▶ Remote application servers wait until network is up again 	Usr: Resolve network problem Usr: Restart transactions
TCP/IP on central instance	Central instance fails (see failure of central instance)	SA: Restart TCP/IP SA: Restart central instance Usr: Restart transactions
TCP/IP on application server	Application server fails (see failure of application server)	SA: Restart TCP/IP SA: Restart application server instance Usr: Restart transactions

Failure	Impact	Actions
TCP/IP on database server	Connection to ICLI server lost (see failure of ICLI server)	SA: Restart TCP/IP Usr: Restart transactions
z/OS LPAR	All components running in the LPAR fail (see failures of individual components)	Usr: Restart of LPAR SA: Restart DB2 SA: Restart other components

2.2.2 Data sharing, sysplex failover, double network (single central instance)

The scenario in Table 2-2 builds on the previous scenario by adding DB2 data sharing, SAP sysplex failover, shared HFS, and a double network with VIPA and OSPF. This scenario is still using the old-style central instance.

Summary: Redundancy and failover capabilities are implemented for database and network. The central instance (inclusive message server) remains a single point of failure.

Table 2-2 DB2 Sysplex Data Sharing configuration with double network

Failure	Impact	Actions
DB2	<ul style="list-style-type: none"> ▶ Rollback of transactions ▶ Local z/OS application servers wait until DB2 is up again ▶ <i>Remote application servers failover to other ICLI servers and DB2 subsystems</i> 	SA: Restart DB2 Usr: Restart transactions
ICLI server	<ul style="list-style-type: none"> ▶ Rollback of transactions ▶ <i>Application servers reconnect to ICLI server or failover to standby ICLI server and DB2 subsystem</i> 	SA: Restart ICLI server Usr: Restart transactions
Central instance	<ul style="list-style-type: none"> ▶ Rollback of transactions ▶ Application servers wait until central instance is up again 	SA: Restart central instance Usr: Restart transactions

Failure	Impact	Actions
Message server	<ul style="list-style-type: none"> ▶ Most transactions are inhibited because the enqueue work process is not reachable ▶ Application servers wait until message server is up again ▶ Group logon is inhibited 	SA: Restart message server Usr: Restart transactions
Application server instance	<ul style="list-style-type: none"> ▶ Transactions on this instance are lost ▶ Rollback of database updates ▶ User sessions on this instance are lost 	Usr: Connect to another instance Usr: Restart transactions SA: Restart instance
Gateway	<ul style="list-style-type: none"> ▶ For most transactions, no impact ▶ Connections to registered RFC servers inhibited until they have reconnected to gateway 	SA: Restart gateway
Syslog collector	<ul style="list-style-type: none"> ▶ For most transactions, no impact ▶ Global syslog file out of date 	SA: Restart syslog collector
saprouter	<ul style="list-style-type: none"> ▶ User sessions lost ▶ Reconnect inhibited 	SA: Restart saprouter Usr: Reconnect
NFS server	<ul style="list-style-type: none"> ▶ Some transactions stop, fail after timeout ▶ Batch transactions stop, fail after timeout ▶ Restart of application servers inhibited ▶ If data was written to file, last written data is in doubt 	SA: Restart NFS server Usr: Restart transactions
File system	<ul style="list-style-type: none"> ▶ For most transactions, no impact ▶ If data was written to file, transaction is rolled back and last written data is in doubt 	Usr: Restart transaction
Network (router, switch, adapter card)	<i>None</i>	None
TCP/IP on central instance	Central instance fails (see failure of central instance)	SA: Restart TCP/IP SA: Restart central instance

Failure	Impact	Actions
TCP/IP on application server	Application server fails (see failure of application server)	SA: Restart TCP/IP SA: Restart application server instance Usr: Restart transactions
TCP/IP on database server	Connection to ICLI server lost (see failure of ICLI server)	SA: Restart TCP/IP Usr: Restart transactions
z/OS LPAR	All components running in the LPAR fail (see failures of individual components)	Usr: Restart of LPAR SA: Restart DB2 SA: Restart other components

2.2.3 Enqueue replication, NFS failover—full-blown HA

The scenario in Table 2-3 builds on the previous two scenarios by adding the new enqueue instance, the enqueue replication server, and NFS failover support. This scenario is the fully implemented SAP high availability solution.

Summary: There is no single point of failure any more. The impact of a failure has a local scope; it is limited to the transactions that are currently using the failing resource. The SAP system remains available.

Table 2-3 Fully implemented SAP high availability solution

Failure	Impact	Actions
DB2	<ul style="list-style-type: none"> ▶ Rollback of transactions ▶ Local application servers wait until DB2 is up again ▶ Remote application servers failover to other ICLI servers and DB2 subsystems 	SA: Restart DB2 Usr: Restart transactions
ICLI server	<ul style="list-style-type: none"> ▶ Rollback of transactions ▶ Reconnect to ICLI server or failover to standby ICLI server and DB2 subsystem 	SA: Restart ICLI server Usr: Restart transactions
Enqueue server	<i>None</i>	SA: Failover enqueue server SA: Move enqueue replication server
Enqueue replication server	<i>None</i>	SA: Restart enqueue replication server

Failure	Impact	Actions
Message server	<ul style="list-style-type: none"> ▶ For most transactions, no impact ▶ Certain transactions inhibited (for example, SM66) ▶ Update/batch workload balancing inhibited ▶ Group logon inhibited 	SA: Restart message server
Application server instance	<ul style="list-style-type: none"> ▶ Transactions on this instance are lost ▶ Rollback of database updates ▶ User sessions on this instance are lost 	Usr: Connect to another instance Usr: Restart transactions SA: Restart instance
Gateway	<ul style="list-style-type: none"> ▶ For most transactions, no impact ▶ Connections to registered RFC servers inhibited until they have reconnected to the gateway 	SA: Restart gateway
Syslog collector	<ul style="list-style-type: none"> ▶ For most transactions, no impact ▶ Global syslog file out of date 	SA: Restart syslog collector
saprouter	<ul style="list-style-type: none"> ▶ User sessions lost ▶ Reconnect inhibited 	SA: Restart saprouter Usr: Reconnect
NFS server	<ul style="list-style-type: none"> ▶ None ▶ If data was written to file, last written data is in doubt 	SA: Restart NFS server
File system	<ul style="list-style-type: none"> ▶ For most transactions, no impact ▶ If data was written to file, transaction is rolled back and last written data is in doubt 	Usr: Restart transaction
Network (router, switch, adapter card)	None	None
TCP/IP on enqueue instance	Enqueue server, message server, gateway, syslog collector fail (see failures of individual components)	SA: Restart TCP/IP SA: Restart enqueue server, message server, gateway, collector

Failure	Impact	Actions
TCP/IP on application server	Application server fails (see failure of application server)	SA: Restart TCP/IP SA: Restart application server instance Usr: Restart transactions
TCP/IP on database server	Connection to ICLI server lost (see failure of ICLI server)	SA: Restart TCP/IP Usr: Restart transactions
z/OS LPAR	All components running in the LPAR fail (see failures of individual components)	Usr: Restart of LPAR SA: Restart DB2 SA: Restart other components

Planning and preparing for the SAP high availability solution

This chapter describes planning tasks to be performed in order to prepare a new, or enable an existing, SAP R/3 on DB2 UDB for OS/390 and z/OS system for the SAP high availability solution using System Automation for OS/390. We accomplish this by describing our high availability configuration and documenting our planning decisions.

The chapter includes the following sections:

- ▶ The SAP High Availability solution configuration
- ▶ Naming conventions
- ▶ Network configuration and setup
- ▶ File system setup
- ▶ DB2 setup
- ▶ ICL server setup
- ▶ Netview and System Automation basic setup
- ▶ SAP installation

3.1 SAP high availability solution configuration

We designed a configuration to demonstrate how System Automation for OS/390 can be used to make all of the necessary SAP components highly available. Our configuration included two LPARs running z/OS in a sysplex with a DB2 data sharing database, and one LPAR with z/VM having Linux guests. This is shown in Figure 3-1.

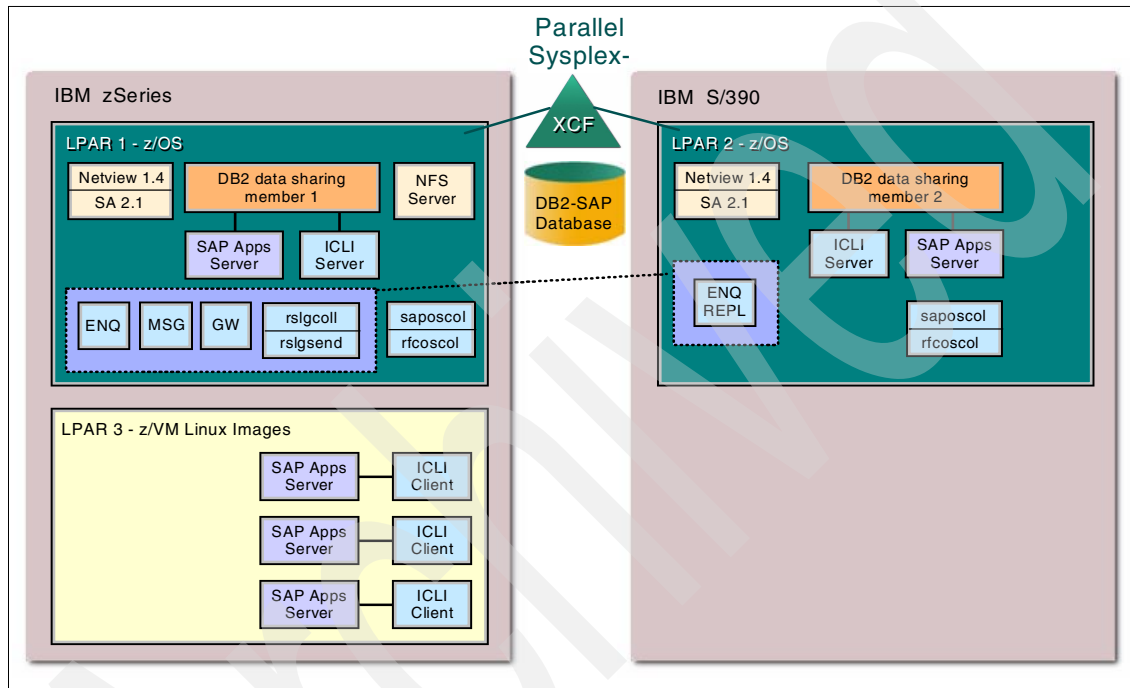


Figure 3-1 SAP high availability solution configuration

3.1.1 Hardware requirements

The next planning task is to decide on the hardware that is needed. Here are the things we need to know to help us make that decision:

- ▶ The new enqueue server and enqueue replication server run on z/OS 1.2, so we need at least two LPARs capable of running z/OS 1.2.
- ▶ The new enqueue server runs on the primary LPAR server with the enqueue replication server running simultaneously on the secondary LPAR. The two servers must access common SAP directories, so shared HFS is required.
- ▶ Shared HFS requires Parallel Sysplex.

- ▶ You can configure multiple LPARs on one physical mainframe server, but a failure of the mainframe server would bring down the entire SAP system. So for production SAP systems we recommend that you use two separate mainframe servers to eliminate the mainframe server as a single point of failure. We chose to have two separate mainframe servers for our test configuration.

Using one or more mainframe servers for non-production SAP systems depends on your business and operating requirements. If you are interested in implementing the SAP high availability solution, but have not ordered or received the hardware you require, you could implement the SAP high availability solution using two LPARs on one mainframe server.

- ▶ DB2 data sharing is a feature that helps make the SAP system highly available by eliminating the database as a single point of failure. We include DB2 data sharing in our solution. DB2 data sharing requires Parallel Sysplex also.
- ▶ We also chose to run a remote application server on Linux on zSeries to show how System Automation for OS/390 can manage remote servers. Therefore, at least one of our mainframe servers must be a zSeries server, because SAP application servers on Linux on zSeries require 64-bit architecture.
- ▶ Since our configuration includes remote application servers, we should consider the network connecting them to the enqueue server. We don't want the network to be a single point of failure. We prevent this by having multiple paths between the enqueue server and the remote application servers. We use the HiperSockets feature on the zSeries server for primary connectivity and use OSA-Express Fast Ethernet as our secondary connectivity.

So our SAP high availability solution configuration requires the following hardware:

- ▶ One zSeries server that supports:
 - z/OS LPAR
 - HiperSockets
 - OSA-Express Fast Ethernet
 - z/VM LPAR that supports Linux for zSeries
 - XCF Links
- ▶ One S/390 server that supports
 - z/OS LPAR
 - OSA-Express Fast Ethernet
 - XCF Links

3.1.2 Software requirements

Table 3-1 summarizes the software requirements. We provide the minimum level of software needed to implement SAP High Availability solution, the actual level of software implemented in the ITSO test environment, and any special PTF requirements for each product. Because the EI is not generally available, be sure to check SAP Note 81737 for the latest PTF requirements.

Table 3-1 Software requirements for the HA solution

Product name	Minimum level requirement	Actual level used at ITSO	PTF requirements
z/OS	V1.2	V1.2	APAR 53313 for the NFS server
DB2 Universal Database for OS/390 and z/OS	Version 6	Version 7	
Tivoli NetView for OS/390 ^a	V1.3	V1.4	
System Automation for OS/390	V2.1	V2.1	APAR OW48503 APAR OW51676
SAP R/3	SAP R/3 4.6B database 4.6D kernel w/ patch level 1145	SAP R/3 4.6C database 4.6D kernel w/ patch level 1145	New enqueue svr New enqueue replication server See SAP note 524816

a. Tivoli NetView is required by System Automation for OS/390

Application server instances were installed on z/OS and Linux on zSeries. Table 3-2 lists the software used for the application server on Linux.

Table 3-2 SAP application server for Linux on zSeries

Product name	Minimum level requirement	Actual level used at ITSO	PTF requirements
Linux on zSeries	SuSE Linux Enterprise Server 7 for zSeries (64 bit)	SuSE Linux Enterprise Server 7 for zSeries (64 bit)	
ICLI	APAR OW53950	APAR OW53950	UW88745
z/VM (optional)	V4.2	V4.2	

Integrated Command Level Interface (ICLI)

For ICLI support of SAP for Linux on zSeries, APAR OW53950 is required. This APAR must also be installed for SAP systems that don't use the ICLI server, because the application server on z/OS requires the ASCII/EBCDIC converter and it is delivered from IBM via the ICLI PTF. See *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966 for details.

3.2 Naming conventions

SAP recommends running one SAP system on one server. However, one of the strengths of z/OS is the capability of running *multiple* SAP systems on one server. One possible configuration is to run all production SAP systems on one server or Parallel Sysplex and run all non-production SAP systems on another server or Parallel Sysplex. In this hypothetical configuration, each SAP system would normally consist of, among other things, its own DB2 subsystem, its own set of file systems, a large number of SMS VSAM data sets, and its own set of ICLI servers. Some common questions that need answers include:

- ▶ How do you monitor all SAP related address spaces with SDSF?
- ▶ On what volumes should I allocate my SMS storage groups?
- ▶ How do I use Work Load Manager (WLM) to prioritize one SAP system over another?

When you consider the number of SAP systems that can run on one server and the management requirements for those SAP systems, it becomes increasingly clear that a good naming convention will make the monitoring and maintenance tasks of each SAP system easier.

An SAP system setup for the SAP high availability solution is also capable of running on a server hosting other SAP systems. The only differences are that there are more components to consider when planning their names. Of course, you could define multiple HA SAP systems in one server or Parallel Sysplex.

How many NFS servers do you need? If you separate the SAP systems and assign two or more LPARs to each SAP system, you probably want to configure multiple NFS servers, one per SAP system so that a failure of one NFS server does not affect other SAP systems. If you run several SAP systems on the same set of LPARs, it is sufficient to have one NFS server that serves multiple SAP systems.

Another reason for multiple NFS servers is that the SAF security option is not useful for server-to-server communication. Instead, you need the export file to provide standard UNIX security. However, if the number of available LPARs is limited and you consider to export user file systems, you may choose to run

multiple NFS servers on the same LPAR. One NFS server exports only the global SAP directories to a list of UNIX servers. The second NFS server uses SAF security to export user directories and let users authorize with their password. You need one TCP/IP stack per NFS server.

So let's get to the task at hand—choosing names for the components of one SAP system. We recommend that you use the 3-character SAP system identification <SID> as a prefix in the name of all the components for one SAP system wherever possible. We recommend using SAP as a prefix for all SAP resources not related to a specific SAP system. In Table 3-3 we list all the components for our SAP system, RED, along with the name we recommend and how or where you define them.

Table 3-3 Naming conventions for single SAP system components

Component	Recommended name	Our name	How/where defined
Jobname for enqueue server	<SID>ADMES	REDADMES	System Automation (SA) definition
Jobname for enq. replication server	<SID>ADMER	REDADMER	SA definition
Jobname for message server	<SID>ADMMS	REDADMMS	SA definition
Jobname for gateway	<SID>ADMGW	REDADMGW	SA definition
Jobname for syslog collector	<SID>ADMCO	REDADMCO	SA definition
Jobname for syslog sender	<SID>ADMSE	REDADMSE	SA definition
Jobname(s) for rfcocol	<SID>ADMRx	REDADMR1	SA definition
Jobname for sapocol	SAPOSCOL	SAPOSCOL	SA definition
Jobname for saprouter	SAPROUTE	SAPROUTE	SA definition
Jobnames for application server instances and their monitors	<SID>ADMnn	APPSRVnn	SA definition

Component	Recommended name	Our name	How/where defined
DB2 address spaces	<SID>xMSTR, <SID>xDBM1, <SID>xIRLM, <SID>xSPAS where x defines the data sharing member	D7XxMSTR, D7XxDBM1, D7XxIRLM, D7XxSPAS where x defines the data sharing member	PROCLIB member names
High Level Qualifier for SAP VSAM objects	<SID>SAP	SAPRED	IDCAMS
High Level Qualifier for Shared HFS file systems	<SID>SHFS.<inst ance-name>	SAPRED.SHFS	MOUNT FILESYSTEM command
ICLI server procedure names	<SID>ICLIx	REDICLIx	PROCLIB member
WLM definitions for service classes	<SID>HIGH, <SID>MED, <SID>LOW	SAPHIGH, SAPMED, SAPLOW	WLM ISPF panels
SMB Share Names	<SID>MNT, <SID>USR, SAPTRANS	SAPMNT, SAPUSR, SAPTRAN	DFS SMB setup
NFS Server procedure name	<SID>NFS or SAPNFS	MVSNFSSA	PROCLIB member
VIPA name for enqueue instance	sap<sid>	sapred	TCP/IP DNS entry
VIPA name for saprouter	saprout	saprout	TCP/IP DNS entry
VIPA name for NFS server	<sid>nfs or sapnfs	sapnfs	TCP/IP DNS entry

3.3 Network configuration

We designed a network to enable SAP high availability, eliminating the network components as a single point of failure; see Figure 3-2.

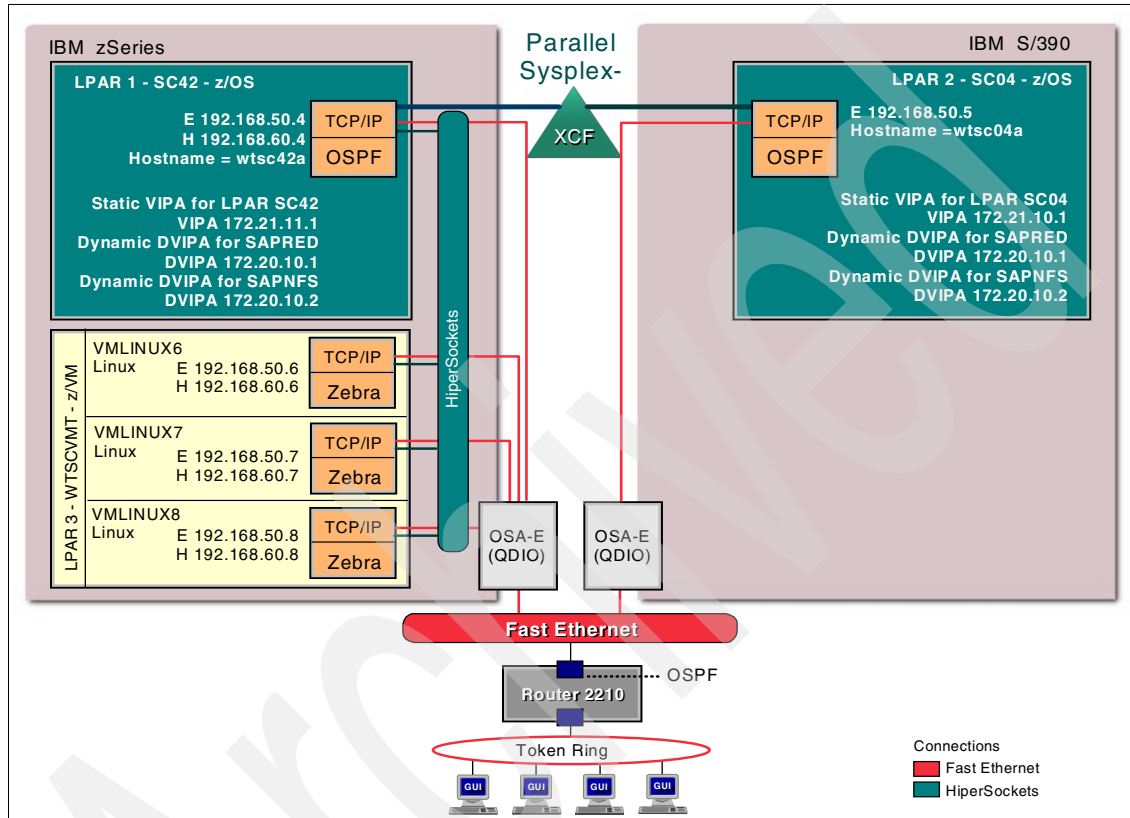


Figure 3-2 SAP high availability solution networking configuration

3.3.1 Network hardware components

We used the following hardware:

- ▶ OSA-Express Fast Ethernet adapter
- ▶ HiperSockets
- ▶ Router

The OSA-Express Fast Ethernet adapter and HiperSockets give us more than one path between the remote application servers and the database servers.

The router is necessary to connect our AIX system to the z/OS LPARs. We used the AIX machine to access the SAP code. We recommend, but did not have, two routers for redundancy.

3.3.2 Networking software components

We prepared software components for z/OS, Linux on zSeries, AIX, and Windows 2000.

z/OS networking software

We implemented the following:

- ▶ TCP/IP
 - Multiple stacks discussion
 - Static virtual IP address (VIPA) definitions for:
 - DB2
 - ICL1 servers
 - Dynamic VIPA definitions for:
 - Enqueue instance
 - NFS server and/or DFS SMB
 - SAP network interface router (saprouter)
 - OSPF
 - NFS Client

TCP/IP

In each of our LPARs, several TCP/IP stacks were already configured. They were configured with common INET (CINET). Stacks configured with Integrated INET perform better than CINET. Therefore, we recommend Integrated INET over CINET.

We defined another stack to isolate our configuration from the existing stacks. The started procedure is documented in Appendix A, “Network setup” on page 217. We had some problems starting our NFS server. We discovered that we needed to configure a portmapper just for our stack. We created a procedure, PMAPA, for our portmapper (see “Portmapper started task” on page 222). This procedure is used by both LPARs. For completeness, we also created a separate procedure, FTPDA, for our FTPD (see “FTP daemon started task” on page 222). The experience taught us that configuring multiple stacks is complex and error prone. We recommend that you use only one TCP/IP stack, if possible.

We had to define a file, `/etc/resolv.conf`, on each LPAR. Otherwise the command `nslookup` ran against the wrong TCP/IP stack. The `/etc/resolv.conf` file for SC04 can be found in “File `/etc/resolv.conf` - SC04” on page 228; for SC42 it can be found in “File `/etc/resolv.conf` - SC42” on page 223.

Network definitions are required in the BPXPRMxx member of SYS1.PARMLIB to support all of the stacks as well as the NFS Client. See “z/OS UNIX System Services setup - BPXPRMxx” on page 217.

We defined a TCP/IP stack, TCPIPA, on both SC42 and SC04. The TCP/IP profile for each stack can be found in “TCPIP profile - SC42” on page 223 and “TCP/IP profile - SC04” on page 228.

In the TCP/IP profile we defined a static VIPA for our hosts containing all the components that we did not need to move, that is, the DB2 subsystem, ICLI servers, and local z/OS dialog instances.

► LPAR SC42

The static VIPA for SC42 in Figure 3-2 on page 40 is 172.21.11.1 and we associated that with hostname wtsc42a. Our primary path on SC42 uses HiperSockets with IP address 192.168.60.4. The secondary path uses OSA-Express Fast Ethernet with address 192.168.50.4.

► LPAR SC04

The static VIPA for SC04 in Figure 3-2 on page 40 is 172.21.10.1 and we associated that with hostname wtsc04a. Our primary path uses OSA-Express Fast Ethernet with address 192.168.50.5. The secondary path uses the XCF link.

In the TCP/IP profile we defined dynamic VIPA for the applications that exist once and only once in the configuration. These applications are the enqueue instance, the NFS server, and the SAP network interface router (saprouter). These applications are moved between our z/OS hosts when necessary by System Automation for OS/390. We could have defined a dynamic VIPA and name for each ICLI server and customized SA to move the ICLI server between LPARs, but we did not. This would make it more difficult for the SAP basis administrator to determine which database server the remote application server is connected to at any one point in time.

We defined a dynamic VIPA for each movable application and assigned a hostname to it. The dynamic VIPA is to be defined as VIPARANGE with the attributes MOVEABLE and DISRUPTIVE:

```
VIPADYNAMIC
VIPARANGE DEFINE MOVEABLE DISRUPTIVE 255.255.255.0 172.20.10.0
ENDVIPADYNAMIC
```

Furthermore, the SOURCEVIPA attribute is needed. Our TCP/IP profile is listed in “TCP/IP profile - SC04” on page 228.

► Enqueue instance

We assigned the name *sapred* to the enqueue instance with a dynamic VIPA of 172.20.10.1 and added it to the DNS.

The actual name assignment is handled by SA. SA calls the procedure and substitutes the IP address for the variable &VIPA. See the following PROCLIB member:

```
***** Top of Data *****
//TCPVIPA PROC  VIPA='0.0.0.0'
//VIPA00 EXEC   PGM=EZBXFDVP,
//              PARM='POSIX(ON) ALL31(ON)/-p TCPIPA -c &VIPA'
***** Bottom of Data *****
```

► NFS server

We assigned the name *sapnfs* to the NFS server with a dynamic VIPA of 172.20.10.2 and added it to the DNS.

The IP address moves to the system where the NFS server is started by coding the exec statement labeled DEFVIPA in B.1, “NFS server procedure” on page 236.

Security considerations: You may have concerns about the attribute SECURITY(EXPORTS). This attribute means that normal UNIX security applies. First of all, the export list of the movable NFS server can be limited to the mentioned global SAP directories, which do not contain sensitive data. Furthermore, the access can be restricted to specific client IP addresses. For further information on setting up NFS, see *Network File System Customization and Operation*, SC26-7417.

► SAP network interface router (saprouter)

We refer to the SAP router as *saprout* with an IP address of 172.20.10.3

The actual assignment is handled again by SA. SA calls the procedure and substitutes the IP address for the variable &VIPA. See the following PROCLIB member:

```
***** Top of Data *****
//TCPVIPA PROC  VIPA='0.0.0.0'
//VIPA00 EXEC   PGM=EZBXFDVP,
//              PARM='POSIX(ON) ALL31(ON)/-p TCPIPA -c &VIPA'
***** Bottom of Data *****
```

It is not required to run saprouter on the same system as the production system. For security reasons, customers might run it on a server associated with a development instance, or they may assign it to a separate TCP/IP stack.

OMPROUTE (OSPF protocol)

OSPF works with static VIPA to permit dynamic routing over two or more paths. We defined a procedure called OMPROUTA that is used by each LPAR. The name matches with the TCP/IP procedure TCPIPA. The procedure for SC42 and SC04 is in “OMPROUTE started task - SC42” on page 226.

As you can see from the procedure, it is a USS program. The OMPROUTE program takes its input from TCPIPA.&SYSNAME.OMPROUTA.ENVVARS, which points to TCPIPA.&SYSNAME.TCPPARMS(TCPDATA) and TCPIPA.&SYSNAME.TCPPARMS(OMPROUTA). The latter file contains our routing definitions.

The SC42 ENVVARS file can be found in “ENVVARS - SC42” on page 227. The routing definitions for LPAR SC42 can be found in “ENVVARS - SC42” on page 227.

The SC04 ENVVARS file can be found in “ENVVARS - SC04” on page 231, and the routing definitions for LPAR SC04 can be found in “ENVVARS - SC04” on page 231

The important thing to note about the ENVVARS file is that statement:

```
RESOLVER_CONFIG=/'TCPIPA.&SYSNAME..TCPPARMS(TCPDATA)'
```

has been overridden by the file /etc/resolv.conf in each LPAR

The important things to note about the routing definitions are:

- ▶ Each possible interface should be defined with the proper MTU size, because the default MTU is 576 for a route that is not in the routing file.
- ▶ The order of the definitions must match the order of the IP addresses in the TCP/IP profile HOME statement.

NFS Client

SAP delivers the R/3 4.6D SR2 product and installation code on CD-ROM. Since z/Series has no ability to attach a CD-ROM directly, we configure NFS Client on z/OS to mount the CD-ROM exported from any UNIX system. We define the NFS Client started task MVSNFSCA. The NFS Client is shown in B.2, “NFS client procedure” on page 238.

Linux on zSeries

Our network configuration diagram shows three Linux on zSeries machines. For the SAP high availability solution we used only vmlinux6. The other two Linux on zSeries images were used by the team writing the redbook *SAP on DB2 UDB for OS/390 and z/OS - Implementing Application Servers for Linux on zSeries*, SG24-6847. In addition to the normal TCP/IP setup, which we don't document in

this redbook, we set up the definitions for two network devices, HiperSockets and OSA-Express. We also set up the Zebra router daemon for dynamic routing. Zebra is the functional equivalent of OMROUTE on z/OS and GATED on AIX. The network interfaces are set up as follows:

modules.conf

To be sure that the correct modules will be loaded by the kernel, we checked that the HiperSockets (hsi0) and OSA-Express (eth1) have one entry in /etc/modules.conf (Figure 3-3).

```
# Configuration file for loadable modules; used by modprobe and kld
#

# Aliases - specify your hardware
alias eth0 lcs
alias hsi0 qeth
alias eth1 qeth
alias tr0 off
alias scsi_hostadapter off
alias fb0 off
```

Figure 3-3 Checking modules.conf

chandev.conf

The next step is to update the file chandev.conf. This file is used to provide the Linux kernel with the parameters related to the device channel layer. We have to make appropriate entries for our two devices.

The HiperSockets device uses the qeth device driver. The addresses for this device are 0x7200,0x7201,0x7202. We add the following line to the end of /etc/chandev.conf file:

```
qeth0,0x7200,0x7201,0x7202
```

To the Fast Ethernet adapter, which also uses the qeth device driver, we add the following lines:

```
add_parms,0x10,0x2880,0x2881,0x2882,portname:OSA2880,secondary_router
qeth1,0x2880,0x2881,0x2882
```

Now we activate our changes to chandev.conf with the command:

```
rcchandev reload
```

For more detailed information concerning the device driver for Linux on zSeries, see *Linux for zSeries: Device Drivers and Installation Commands (March 4,2002)*, Linux Kernel 2.4, LNUX-1103.

We assigned the HiperSockets device an IP address of 192.168.50.6 and assigned the OSA-Express device an IP address of 192.168.60.6.

Zebra

The steps to set up Zebra are as follows:

- ▶ Create the file `ospfd.conf` in directory `/etc/zebra`. This file is used by `ospfd` to give routing instructions to the Zebra routing daemon. We created the file from the sample file `ospfd.conf.sample`. The contents of this file are in A.2.1, “Zebra setup - OSPF” on page 233.
- ▶ Create the file `zebra.conf` in directory `/etc/zebra`. This file is used by Zebra to perform the actual routing of the sessions. We created the file from the sample file `zebra.conf.sample`. The contents of this file are in A.2.2, “Zebra setup - Zebra” on page 233
- ▶ Verify that the start parameters in file `/etc/rc.config` include starting both the OSPF daemon and zebra daemon at boot time. For example:

`START_OSPFD="yes" and START_ZEBRA="yes"`

To start dynamic routing dynamically, use the commands:

`/etc/init.d/zebra start`

`/etc/init.d/ospfd start`

AIX

We implemented the following to give us access to the SAP code on CD-ROM:

- ▶ TCP/IP
- ▶ NFS server

We do not give the details here; find them in *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966.

Windows 2000

To share z/OS files with Windows 2000 files, DFS SMB needs to be installed and configured on the z/OS platform. See *Distributed File Service SMB Administration*, SC24-5918.

3.4 File system setup

Shared HFS is required to allow the failover of the SAP instances. Furthermore, it is needed for the movable NFS server.

The Shared HFS feature allows you to define shared as well as system-specific file systems, by using special variables in the path name. If you have all your SAP

systems within a sysplex, you can share all the files. If you, for example, have one production sysplex and one test sysplex, and still want to use the same file systems (for example, the Transport directory), you must use the NFS Server/Client feature. NFS Server must run on the system that owns the directory, and NFS Client must run on the other system.

File systems

We recommend that the non-z/OS executables and profiles be stored in a central location; we chose z/OS for that location. Therefore, we required that NFS Server or DFS/SMB be set up on z/OS, and the SAP file systems on the z/OS shared file systems be exported or shared to the non-z/OS hosts.

The SAP profiles for each application server are stored in the same directory with different names, so we export ed just one directory to all non-z/OS application servers.

The executables have the same name for all platforms so you have to create specific executable directories in addition to the standard executable directory `sapmnt/<sid>/exe`. For our configuration we defined the following directory:

For Linux: `/sapmnt/RED/Linux/exe`

Here are some suggestions for other application server environments:

For AIX: `/sapmnt/RED/AIX/exe` (not defined, but supplied as example)

For Windows 2000: `/sapmnt/RED/W2K/exe` (not defined, but supplied as example)

For Sun Solaris: `/sapmnt/RED/Solaris/exe` (not defined, but supplied as example)

Actually, to be consistent, it would make sense to create a directory for USS or z/OS: `/sapmnt/RED/zOS/exe` (we did not do this)

We can think of two ways to mount the file systems:

- ▶ Mount these file systems on the NFS or DFS/SMB client side (non-z/OS side) with the same name. The advantage we see in doing this is that it makes it easier for the system programmer to distinguish the executable directory no matter where he/she is logged on.
- ▶ Mount the file systems on the non-z/OS side with the names that SAP expects. This has the advantage of not requiring an additional UNIX soft link.

We chose the first method. R3SETUP needs to find these directories, so we created soft links from the standard SAP names to the mounted file systems.

For example, in directory `/sapmnt/RED/` on a non-z/OS host, we executed the command:

```
ln -s /sapmnt/RED/Linux/exe exe
```

Figure 3-4 shows the SAP directory structure and file systems for the enqueue instance. This is similar to the old central instance except that the instance name is different.

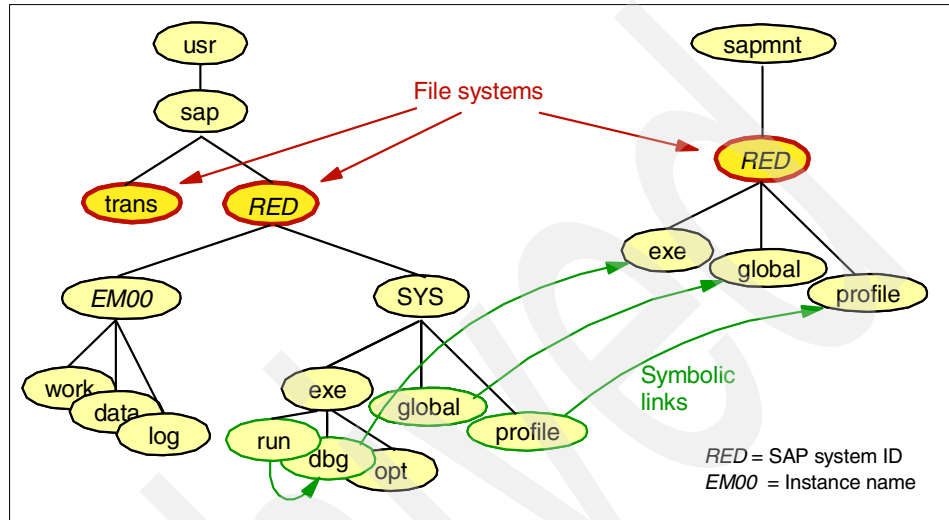


Figure 3-4 Directory tree

All SAP directories and file systems have to be defined as follows.

3.4.1 SAP directory definitions

The following directories must be defined:

SAP global transport directory

The directory `/usr/sap/trans` must be globally accessible and needs to be shared. In addition, it needs to be exported by the NFS server.

SAP system-wide directories

The subdirectories of `/usr/sap/<SAPSID>/SYS` are usually defined at installation time as symbolic links to the corresponding subdirectories of `/sapmnt/<SAPSID>`, for example, `/usr/sap/RED/SYS/profile` points to `/sapmnt/RED/profile`. The directory `/sapmnt` is to be created in the root file system and thereby shared in the sysplex.

The directory /sapmnt/RED is the mount point for the SAP system-wide file system. This file system needs to be exported by the NFS server such that it can be mounted by remote application server instances.

SAP local directories

On z/OS the directory /usr is a symbolic link to \$SYSNAME/usr . That means that the contents of the /usr directory is different on every LPAR. This, however, is not practical for the /usr/sap directory. We propose to create the directory /sap in the root file system and to define symbolic links for /usr/sap to point to /sap. The symbolic links must be defined on each LPAR, i. e. in each \$SYSNAME/usr. With this approach the subdirectories of /usr/sap are identical on all z/OS systems.

The /sap (alias /usr/sap) directory contains the mount points /usr/sap/<SAPSID> for the instance-specific file systems. Those file systems do not need to be exported by NFS.

There is also a /usr/sap/tmp directory. For performance reason, this should not be shared across the Sysplex. Define it as symbolic link to /tmp (which points to \$SYSNAME/tmp).

In a shared HFS environment, the file system should be mounted and owned by the LPAR where the instance runs. One reason is performance, the other is to isolate the impact of an LPAR failure to the failing LPAR. (If you allow the instance directory to be owned by a different LPAR, a failure of this LPAR causes the application server to loose access to open files. This would require a restart of the application server.)

If you run multiple instances on z/OS belonging to the same SAP system, for example the enqueue instance and additional dialog instances, we recommend to setup multiple file systems. In our case, we run the enqueue instance EM00 and the dialog instances D10 and D11. Two additional file systems should be created and mounted on /usr/sap/RED/D10 and /usr/sap/RED/D11. While the ownership of /usr/sap/RED is moved with the enqueue instance /usr/sap/RED/D10 and /usr/sap/RED/D11 are always owned by the LPAR the instance is configured for. This ensures optimal separation and performance.

Administrator's home directory

The home directory <sapsid>adm is shared in the sysplex.

SAPOSCOL/RFCOSCOL directory

If you have different versions of SAP R/3 on the same system, the executables and the startup scripts of SAPOSCOL and RFCOSCOL should be placed in their own directory. This should be a shared directory, for example /usr/sap/saposcol.

SAP executables directory - Linux only

We also put the SAP executables of the dialog instance on Linux in the Shared HFS, under /sapmnt/<SAPSID>/Linux/exe. If you plan to do this, you will have some challenges to solve manually. These are about the user ID on the z/OS server side. Since user ID *root* is used when running the SAP installation utility R3setup on Linux, files are mapped to UID and GID 65534 (or -2, which means *nobody*), when the files reside on z/OS. Some manual steps must then be done to get the correct UID and GID. R3setup then has to be restarted. These steps are described in detail in *SAP on DB2 UDB for OS/390 and z/OS - Implementing Application Servers for Linux on zSeries*, SG24-6847.

Alternatively, install the executables in a local directory first, and then copy them into the HFS directory later.

The SAP directories we used are shown in Table 3-4.

Table 3-4 SAP directories

Description of directory name	Directory name
Administrator's home directory	/u/redadm
SAP directory tree	/usr/sap
Global transport directory	/usr/sap/trans
SAP system-wide directory tree	/sapmnt/RED /usr/sap/RED/SYS
Directory tree for enqueue instance	/usr/sap/RED/EM00
Directory for Linux executables	/sapmnt/RED/Linux/exe

Manually added links are shown in Table 3-5.

Table 3-5 Manually added links

Manually added links	
SAP directory tree in the Shared HFS	/usr/sap --> /sap /usr/sap/tmp --> /tmp
On remote application server (here Linux)	/sapmnt/RED/exe --> /sapmnt/RED/Linux/exe

Table 3-6 on page 51 shows the symbolic links created automatically during SAP installation.

Table 3-6 Symbolic links created by R3setup

Symbolic links created during SAP installation

SAP system directory tree:

- /usr/sap/RED/SYS/global --> /sapmnt/RED/global
- /usr/sap/RED/SYS/profile --> /sapmnt/RED/profile
- /usr/sap/RED/SYS/exe/run --> /usr/sap/RED/SYS/exe/dbg
- /usr/sap/RED/SYS/exe/dbg --> /sapmnt/RED/exe

SAP file system definitions

The Shared HFS data sets are mounted by submitting the JCL shown in Figure 3-5.

```
//SAPRESIM JOB (999,P0K),CLASS=A,MSGCLASS=T,  
//          NOTIFY=&SYSUID,TIME=1440,REGION=0M  
//IKJEFT01 EXEC PGM=IKJEFT1A  
//SYSTSPRT DD SYSOUT=*  
//SYSTSIN DD *  
    MOUNT FILESYSTEM('SAPRED.SHFS.SAPMNT')  
    MOUNTPPOINT('/sapmnt/RED') +  
    TYPE(HFS) MODE(RDWR)  
    MOUNT FILESYSTEM('SAPRED.SHFS.SAPUSR')  
    MOUNTPPOINT('/sap/RED')  
    TYPE(HFS) MODE(RDWR)  
    MOUNT FILESYSTEM('SAPRED.SHFS.TRANS')  
    MOUNTPPOINT('/sap/trans')  
    TYPE(HFS) MODE(RDWR)
```

Figure 3-5 Mount file system commands

An example of mounting the shared HFS from Linux is as follows:

```
sapnfs:/hfs/sapmnt/RED/Linux/exe,text,xlat(oemvs311) /sapmnt/RED/Linux/exe nfs
intr,rsize=8192,wsiz=8192
```

This should be placed in `/etc/fstab` on the Linux server for automatic mount, during restart. An example of this is in B.3, “Mount commands on Linux `/etc/fstab`” on page 238.

Sizing considerations

When you decide to use shared HFS, the sizing considerations in *R/3 Installation on OS/390 Unix System Services*, Release 4.6C SR2, have to be adapted.

- ▶ SAPMNT should have a primary allocation of a minimum of 500 MB, or CYL=600 (3390-3). This is sufficient to store the executables for two platforms. When using more than two platforms, this will increase.
- ▶ SAPUSR should have a primary allocation of a minimum of 150 MB, or CYL=200 (3390-3). For more than two instances, this will increase.
- ▶ SAPTRANS has a minimum requirement of 100 MB, or CYL=140 (3390-3), per instance. This may become much larger, depending on your environment and the change activity.

3.5 DB2 setup

The processes of the new enqueue instance (EI) do not connect to a database server, so there is no technical requirement to install an EI in the same LPAR as a DB2 subsystem. However, it doesn't make sense to make the EI highly available without making the database server highly available also. So we decided to implement DB2 data sharing.

3.5.1 DB2 data sharing

To prepare for the EI we installed a CI on LPAR SC42 and two dialog instances; one per LPAR. The old central instance or dialog instance running on z/OS can only connect to a DB2 subsystem in the same z/OS LPAR as the central instance or dialog instance. Therefore, our HA solution requires a DB2 data sharing environment. The system configuration we set up consisted of a 2-way data sharing environment. We were not able to follow our own naming conventions. We had to conform to the existing ITSO conventions. So we created one member, D7X1, to run on z/OS LPAR SC42. We created a second member, D7X2, to run on SC04. We defined the group attach name as D7XG. We installed the DB2 modules in load library DB7X7.SDSNLOAD. The ZPARM for both D7X1 and D7X2 are shown in C.1, "DB2 ZPARMS" on page 242. The DB2 ZPARM shows the latest recommended values for maximizing SAP performance.

In addition to the normal JCL jobs to install a DB2 subsystem, we had to customize and run job DSNTIJEX. This enables DB2 secondary authorization exits, which allows you to connect the database with userid <sid>adm, which is attached to group SAPR3. SAPR3 is the owner of all SAP objects.

3.5.2 ARM policy

We had the choice to use either Automatic Restart Manager (ARM) or System Automation for OS/390 to automatically restart the DB2 subsystem in case of a

failure. We chose to use ARM to demonstrate the interoperability of ARM and System Automation.

Our ARM policy is shown in C.2, “The ARM policy” on page 255. It is set up according to the following requirements:

- ▶ “Normal” restart in place, if the LPAR is available.
- ▶ “Light” restart on the other LPAR, if the LPAR is not available.

The LIGHT option of the START DB2 command is only available with DB2 UDB for OS/390 and z/OS version 7. It lets you restart a DB2 data sharing member with a minimal storage footprint, and then terminate normally after DB2 frees the retain locks.

For details about ARM and restart light, refer to *DB2 UDB for OS/390 and z/OS Version 7: Data Sharing Planning and Administration*, SC26-9935.

3.6 Integrated Command Level Interface Server (ICLI Server)

We need an ICLI server for connectivity between the remote application server and the database server. Since we are using DB2 data sharing, SA does not have to move the DB2 subsystem. Likewise, we don't have to move the ICLI server. So we need to configure at least one ICLI server on each LPAR that has a DB2 data sharing member.

If you have more than one remote application server that connects to the database server, we recommend that you create one ICLI server listening on a separate port for each remote application server. In the event that any one ICLI server experiences a failure, the other remote application servers will continue to operate.

For our configuration, we defined three ICLI servers to run on each LPAR, because we have 3 SAP dialog instances running on 3 Linux on zSeries systems. We coded the same port numbers for the ICLI servers on both LPARs.

We recommend that the userid associated with the ICLI server be the <sid>adm user so that all SAP address spaces have the authority given to <sid>adm by RACF. This is easier to manage since you have only one userid instead of having one userid for each ICLI server. A common userid also gives a performance benefit. DB2 caches SQL statements in the statement cache qualified by the userid. Using a single userid saves valuable memory in the DB2 DBM1 address space by reducing the size needed for the DB2 EDMPOOL.

See “ICLI server started task” on page 222 for one of our three ICLI server started task proclib members.

3.7 Netview and System Automation basic setup

Before you start to customize your System Automation for OS/390 policy for the SAP high availability solution, make sure that the basic installation of NetView and System Automation for OS/390 has been finished.

The following z/OS resources should be defined to System Automation for OS/390:

- APPC
- ASCH
- HSM
- JES
- LLA
- Netview, Netview Subsystem Interface and Netview UNIX Server
- OAM
- OMROUTE
- RMF
- RRS
- SA Automation Manager
- TCP/IP
- TSO
- VLF
- VTAM

The Automated Restart Manager (ARM) configuration needs to be checked to ensure that it does not interfere with System Automation. The only subsystem we use ARM with is DB2, which in case of an abend is restarted “light” for cleanup on a different system.

We found the Status Display Facility (SDF) function of System Automation for OS/390 very useful when it came to moving the SAP components between the LPARs. If you want to use SDF, define an SDF focal point and perhaps an SDF backup focal point on your systems. Of course, if you have the NetView Management Console (NMC) installed, you can use it instead of SDF.

Stop the system and re-IPL it. Make sure that System Automation starts all applications and puts them into a “green” status.

3.8 SAP installation

This section discusses the installation of an SAP R/3 4.6C SR2 system, hereafter known as the SAP system. We discuss two cases:

- ▶ Installing a new SAP system on z/OS
- ▶ Preparing an existing system for the HA solution

If you install a new SAP system (as we did), we recommend the following steps (which are described in detail next in “New SAP system installation”):

- ▶ Set up z/OS for SAP.
 - Define the user IDs.
 - Allocate the file systems.
 - Set up the network.
- ▶ Install the central instance on z/OS.
 - Load the database.
- ▶ Install additional dialog instances on z/OS and/or on remote systems.

If you already have a running SAP system and the central instance is not on z/OS, you might plan to just install the enqueue instance on z/OS and to move the global file systems to the Shared HFS. In this case, perform the following steps, which are described in more detail in 3.8.2, “Preparing an existing SAP system for HA” on page 61:

- ▶ Set up z/OS for SAP.
 - Define the user IDs.
 - Allocate the file systems and move the global SAP directories to z/OS.

If you already have a running SAP system and the central instance is located on z/OS, no further preparation steps with respect to the SAP system are required.

3.8.1 New SAP system installation

It is assumed that the network setup and shared file system setup described in this chapter were completed prior to beginning the SAP system installation. It is also assumed that the reader is familiar with *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966, *SAP R/3 on DB2 for OS/390: Connectivity Guide*, SC33-7966-02, and *R/3 Installation on OS/390 Unix System Services*, Release 4.6C SR2. The Planning Guide and the Connectivity Guide are the best sources for planning the hardware and software needed to install an SAP system. Review OSS Note 387127 for last-minute changes to *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966 and *R/3 Installation on OS/390 Unix System Services*, Release 4.6C SR2.

Define an installation userid on z/OS

Define a USS superuser as described in *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966. Give it DB2 SYSADM access to execute DB2 BIND and GRANT commands. We used userid SAPRES1.

Define the SAP system user

Define a USS user <sid>adm, where <sid> is replaced by the SAP system id. Our <sid> was *red*. We recommend that you assign a unique UID and GID in your enterprise. This makes it easier to handle security requirements if you choose to set up a heterogeneous environment of SAP application servers at a later date.

SAP recommends that the <sid>adm user use the C-shell. The C-shell is delivered with OS/390 starting with version 2 release 9. IBM provides the C-shell as file /bin/tcsh. This is an enhanced but compatible version of the C-shell. When you define the userid to RACF, you should specify the C-shell as the initial program. One program that you could specify is /bin/tcsh. We chose to create a soft link from the IBM-delivered name to the common UNIX name. The command we used was:

```
ln /bin/tcsh /bin/csh
```

Install the central instance

Log on directly to USS running on LPAR SC42 using your favorite telnet or login program. Since we exported the SAP kernel CD-ROM from AIX to z/OS and were sitting next to the AIX box, we used the command:

```
rlogin wtsc42a -l <userid>
```

where userid is a USS super userid with UID 0.

Do not log in using a non-superuser USS userid and then issue the **su** command to switch to USS superuser.

Next, from a z/OS TSO user mount the CD-ROM containing the SAP kernel CD. We used the following commands:

```
MKDIR '/SC42/cdrom' MODE(7 5 5)
MOUNT FILESYSTEM('CDROM')
MOUNTPPOINT('/SC42/cdrom')
PARM('erprisc2:/cdrom,XLAT(Y),VERS(2)')
TYPE(NFS) MODE(READ) nowait
```

Follow the instructions in *R/3 Installation on OS/390 Unix System Services*, Release 4.6C SR2 to install the central instance.

After running /SC42/cdrom/UNIX/INSTTOOL.SH to copy the installation program and scripts to the installation directory, make a copy of the CENTRAL.R3S control file. This step makes it easier to start over in case you make any errors replying to prompts. We suggest the following:

```
cp CENTRAL.R3S ci
```

We used the installation program R3SETUP to install the CI:

```
./R3SETUP -f ci
```

R3SETUP prompts the installer for system configuration information. We do not document all responses in this book. Instead, we document only those very few responses necessary for proper implementation of HA.

When prompted for the hostname for the CI, we specified the hostname associated with the static Virtual IP Address (VIPA) of z/OS image SC42. Our hostname was wtsc42a. It is much easier to enable the SAP high availability solution if you have a hostname associated with static VIPA before you begin the CI install.

At the time of our CI install, our SC42 LPAR had more than one TCP/IP stack running, but none of them had VIPA configured. We installed the CI and had SAP stop and start profiles and the R/3 instance profiles defined with a non-VIPAd hostname. We then decided to define another TCP/IP stack with VIPA to isolate our test environment from the other ITSO projects. As a result, we were faced with two choices, 1) change the hostname string in all the R/3 files, or 2) re-install the CI after the new TCP/IP stack was available. In the interest of saving time, we chose to rename the necessary files. We do not recommend that you proceed this way. It is less error prone to let R3SETUP create the files with the proper names.

When prompted for the hostname of the database instance, we entered the same value, wtsc42a, as for the central instance.

Test the CI installation. Log in to USS using <sid>adm and execute the command **R3trans -x**. All is well if you receive a return code of 0000.

Install the database server instance

After installing the central instance, install the database instance. Follow the instructions in *R/3 Installation on OS/390 Unix System Services*, Release 4.6C SR2 to install the database instance

We made a copy of DATABASE.R3S in case we had to start over. We suggest:

```
cp DATABASE.R3S db
```

Before we started the installation of the database instance, we made sure that R3SETUP had access to the DB2 RRSF interface modules. There are a couple of ways to accomplish this. One way is setting the STEPLIB environment variable to point to the DB2 load library. The other way is by adding the DB2 load library to your equivalent of SYS1.LNKST. We chose to use the STEPLIB method, because the systems we were using had other DB2 subsystems defined and running. We executed the command:

```
export STEPLIB=D7X7U.SDSNEXIT:D7X7.SDSNLOAD
```

We used the installation program R3SETUP again to install the database instance:

```
./R3SETUP -f db
```

R3SETUP prompts the installer for system configuration information. We do not document all responses in this book. One group of prompts asks for the location to copy the export CDs to. R3SETUP requires access to all four export CDs simultaneously. We found that we could not copy just three CDs and access the fourth CD from the NFS-mounted cdrom on erprisc2. Therefore, we copied all four CDs to an HFS on USS.

Install additional z/OS Dialog Instances

In addition to the basic SAP-defined 2-tier configuration, we defined two additional local DIs. The reason was that the original “old” CI with its assortment of work process (WP) types, the WPs used to perform real work, will no longer be used after the SAP high availability solution is implemented. The DIs that we installed on each z/OS LPAR provided the necessary SAP dialog, batch, update, etc., WPs.

Installing additional local DIs in the same z/OS image is optional. We chose to install them to show that the SAP system can be managed by SA.

Dialog Instance D10 on SC42(wtsc42a)

Installing a DI is similar to installing a CI. The steps are:

Log on directly to USS running on hostname wtsc42a (LPAR SC42) using your favorite telnet or login program. For this install, we used a freeware telnet-like product called **putty**. This telnet command is GUI based for the login process.

Follow the instructions in *R/3 Installation on OS/390 Unix System Services*, Release 4.6C SR2 for installing a Dialog Instance.

SAP recommends that you begin with an empty install directory, so we executed the shell script /SC42/cdrom/UNIX/INSTTOOL.SH to copy the installation program and control files from the cdrom. If you like, you can copy the files from the CI install directory and delete all of the log files.

Make a copy of the DIALOG.R3S control file. This step makes it easier to start over in case of an error. It also is easier to install additional DIs without having to install the files from the cdrom again. We suggest:

```
cp DIALOG.R3S di
```

Note: Since a dialog instance is installed on the same LPAR as the central instance, make a backup of the SAP files in directory /\$HOME/<sid>adm. The files are:

- ▶ .sapenv_<hostname>.sh
- ▶ .sapenv_<hostname>.csh
- ▶ .dbenv_<hostname>.sh
- ▶ .dbenv_<hostname>.csh

R3SETUP replaces these files with new files.

Now we are ready to use the installation program R3SETUP to install the DI:

```
./R3SETUP -f did10
```

R3SETUP prompts the installer for system configuration information. Again we do not document all responses in this book. Instead, we document only those very few responses necessary for proper implementation of HA.

The hostname for the DI was wtsc42a. This name was associated with the static Virtual IP Address (VIP) of our TCP/IP stack in z/OS image SC42.

We chose an SAP instance number of 10 to distinguish the DI from the CI and any other DIs that we planned to install into the shared file system. So the installation program creates a dialog instance of D10.

We entered wtsc42a as the hostname for the central instance.

We entered wtsc42a as the hostname for the database instance.

The DIALOG.R3S control file does not contain an EXITJCLJOB CARD step to allow you to examine the jobcard.tpl file so that you can adapt it to your particular environment. The installation fails because the JCL job fails with a JCL error. SAP recommends in OSS 393865 that you modify the FOMEBIND.jcl and FOMEGRNT.jcl files directly if you need to update the jobcard.tpl.

Note: OSS Note 393865 pertains to SAP release 6.10, but we found that it applied to our 4.6C SR2 install also.

When the installation is finished you receive the message R3SETUp finished.

We attempted to start the DI, but it would not start. We investigated and found that redadm's environmental files were overlaid. Actually, they were renamed to contain ".0" appended to the original file name. In addition to not being able to start the DI, we were no longer able to start the CI. This is why it is important to save the environmental files before installing the dialog instance. We added the command

```
setenv STEPLIB D7X7.SDSNEXIT:D7X7.SDSNLOAD
```

to the file .dbenv_wtsc42a.csh file, and the command

```
STEPLIB=D7X7.SDSNEXIT:D7X7.SDSNLOAD; export STEPLIB
```

to the .dbenv_wtsc42a.sh file.

This overlay problem happens only when you install a DI on the same host as your CI.

Dialog Instance D11 on SC04 (wtsc04a)

We chose to install another dialog instance on the LPAR SC04. This allowed us the flexibility to perform SAP transactions on a z/OS server when the primary z/OS server is down and the enqueue server has been moved to SC04.

This section is the same as for D10, except that we specified a different hostname for the database instance—wtsc04a.

We chose to perform the install by logging on to USS from TSO. This made the ISPF lovers happy!

Install additional non-z/OS Dialog Instances

We installed additional remote non-z/OS DIs to demonstrate that SA is capable of managing remote SAP application servers.

Dialog Instance D00 on Linux for zSeries

Log on directly to the Linux system using your favorite telnet or logon command. We used the Putty.exe freeware program.

Follow the instructions in *SAP R/3 Installation on Linux for zSeries: IBM DB2 UDB for OS/390 and z/OS*, Release 4.6C SR2 for installing a Dialog Instance.

We did not have the Generally Available kernel CD from SAP at the time of installing the SAP kernel for Linux for zSeries. We used a softcopy of the CD sent to us by the SAP/IBM porting team. However, it is expected that the installation procedure will not change significantly, if at all, by the time the GA kernel CD is available.

Execute the shell script /cdrom/UNIX/INSTTOOL.SH to copy the installation program and control files from the CD-ROM. Make a copy of the DIALOG.R3S control file. This step makes it easier to start over in case of an error. It is also easier to install additional DIs without having to install the files from the CD-ROM again. We suggest:

```
cp DIALOG.R3S di
```

We used the installation program R3SETUP to install the DI, as follows:

```
./R3SETUP -f di
```

After the installation is complete, set up the sysplex failover support for the database connection as described in “Enabling SYSPLEX Failover support for an application server” in *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966. It is important that the environment variables (for example, RSDB_DB2HOSTSTANDBY) are defined as well. This makes it possible to start the application server even if the primary database server is not available. The reason is that the startsap script checks the database connection using R3trans before a startup of the instance is attempted. R3trans does not use the instance profile; instead, it relies on the environment variables.

3.8.2 Preparing an existing SAP system for HA

In this section we describe a method of extending an existing 3-tier SAP system to include application servers running on z/OS.

Define an installation userid on z/OS

Define a USS superuser as described in *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966. Give it DB2 SYSADM access to execute DB2 BIND and GRANT commands. We used userid SAPRES1. Review OSS Note 387127 for last-minute changes to *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966 and *R/3 Installation on OS/390 Unix System Services*, Release 4.6C SR2.

Define the SAP system user

Define a USS user <sid>adm, where <sid> is replaced by the SAP system ID. Our <sid> was *red*.

SAP recommends that the <sid>adm user use the C-shell. The C-shell is included in OS/390 as of version 2 release 9. IBM provides the C-shell as file /bin/tcsh. We issued the command:

```
ln /bin/tcsh /bin/csh
```

to be more consistent with other UNIX platforms.

The UID and GID must match the UID and GID of the <sid>adm user defined on erprisc2. We defined a user redadm with UID=204 and GID=202.

File systems

Create the file systems on z/OS as described in 3.4, “File system setup” on page 46. Then move the contents of the global SAP directories to the newly created file systems on z/OS. This can be accomplished with the following steps:

- ▶ Stop the SAP system.
- ▶ On the central instance, rename the old global directory, for example:

```
mv /sapmnt/RED /sapmnt/RED.old
```
- ▶ Create the new subdirectories, for example:

```
mkdir /sapmnt/RED
mkdir /sapmnt/RED/global
mkdir /sapmnt/RED/profile
mkdir /sapmnt/RED/Linux
mkdir /sapmnt/RED/Linux/exe
ln -s /sapmnt/RED/Linux/exe /sapmnt/RED/exe
```
- ▶ As superuser, mount the z/OS file systems on the central instance:

```
mount 'sapnfs:/HFS/sapmnt/RED/global,text,xlat(oemvs311)'
/sapmnt/RED/global nfs intr,rsiz=8192,wsiz=8192

mount 'sapnfs:/HFS/sapmnt/RED/profile,text,xlat(oemvs311)'
/sapmnt/RED/profile nfs intr,rsiz=8192,wsiz=8192

mount 'sapnfs:/HFS/sapmnt/RED/Linux/exe,text,xlat(oemvs311)'
/sapmnt/RED/Linux/exe nfs intr,rsiz=8192,wsiz=8192
```
- ▶ Copy the following files:

```
cp -pR /sapmnt/RED.old/global/* /sapmnt/RED/global/
cp -pR /sapmnt/RED.old/profile/* /sapmnt/RED/profile/
cp -pR /sapmnt/RED.old/exe/* /sapmnt/RED/Linux/exe/
```
- ▶ Change the mount commands on all other remote application servers to point to the z/OS system instead of the previous central instance and remount the file systems.
- ▶ Start the SAP system.

3.8.3 SAP license

For normal SAP installations, you must obtain an SAP license for the LPAR where the message server runs. In our configuration, the message server was part of the new enqueue instance. Request an SAP license for each LPAR that

will host the enqueue instance. For the SAP commands, see 4.2.2, “Enqueue instance” on page 73.

3.8.4 SAP logon groups

Tip: We recommend that you define LOGON groups.

LOGON groups are used to automatically distribute user logons to individual instances (application servers) or to groups of SAP instances. They are also useful for reconnection to another SAP instance in case the SAPGUI connection or the instance itself become unavailable.

For the setup of LOGON groups, refer to the OSS note 26317 “Set up LOGON group for autom. load balancing”. We defined a LOGON group called PUBLIC with all three instances. See Figure 3-6.

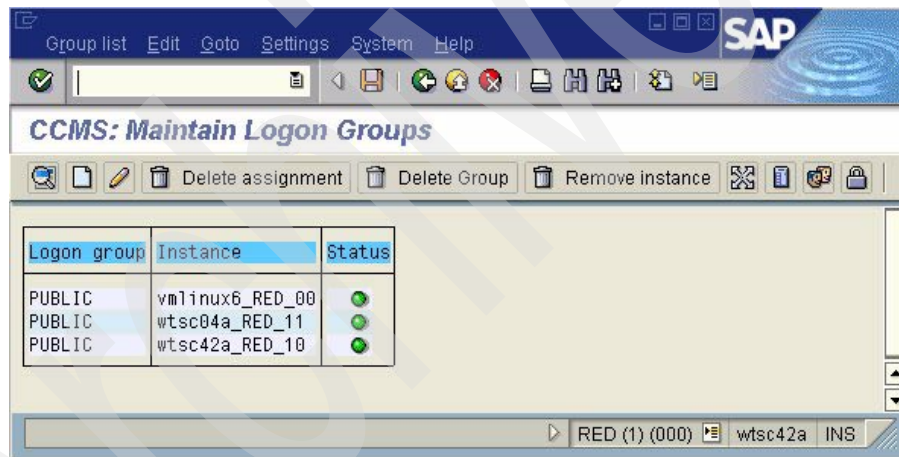


Figure 3-6 LOGON groups

Because the GUI connection has to go to the host where the message server is running, the VIPA of the enqueue instance must be used, as shown in Figure 3-7 on page 64 and Figure 3-8 on page 64.

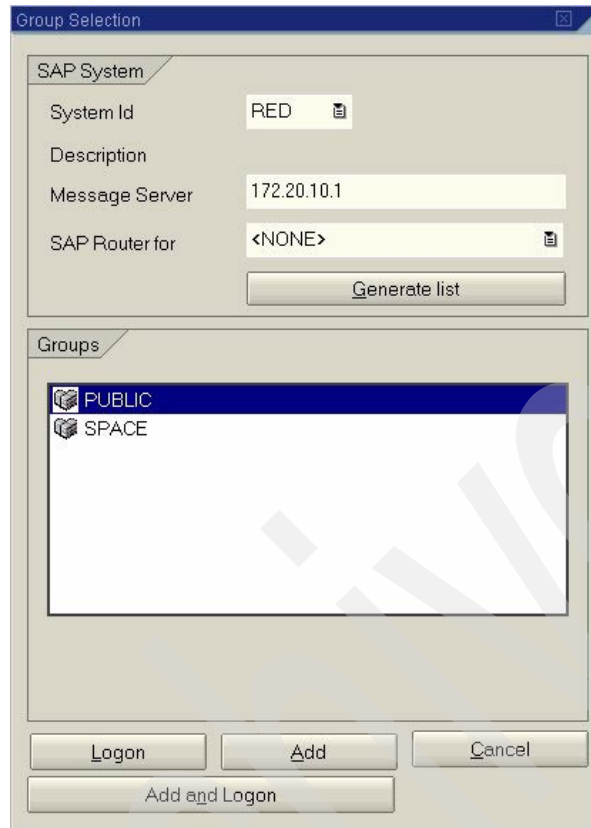


Figure 3-7 Group selection

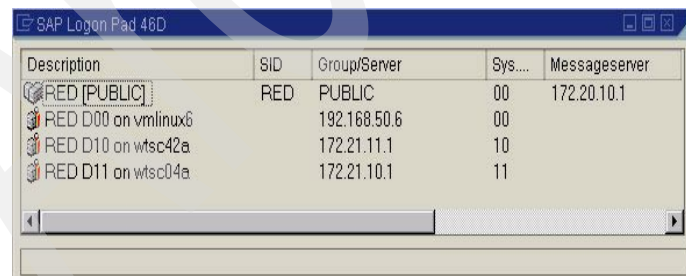


Figure 3-8 SAP logon pad 46D

Customizing SAP for high availability

In this chapter, we describe what you need to do to implement the high availability solution on an existing SAP environment.

The chapter covers the following:

- ▶ How to configure the SAP enqueue instance including the new standalone enqueue server
- ▶ How to configure the SAP environment for System Automation

4.1 Installing and configuring the enqueue instance

Before you start installing the enqueue instance, you must have a running SAP system.

Preferably, you should allocate the file systems needed by SAP on z/OS, and install the central instance on z/OS.

If you do not install the central instance on z/OS, then you should at a minimum perform the installation steps for one dialog instance. This ensures that the parameters for UNIX System Services are appropriate for SAP; the sap<sapsid> user environment is defined; and the standard SAP directory structure is created.

For details about preparation, refer to 3.8, “SAP installation” on page 55.

4.1.1 Getting the standalone enqueue server code from SAP

At the time of writing, the standalone enqueue server has limited availability and is distributed by SAP on request.

It will soon be made available on the SAP Service Marketplace, and you'll be able to download it as a binary patch under section SAP WEB AS 6.20. The enqueue server is compatible with kernel release 4.6D (refer to SAP note 524816). Beginning with kernel release 6.30, it will be part of the normal delivery.

The following parts are new:

enserver	Standalone enqueue server
enrepserver	Enqueue replication server
ensmon	Enqueue monitor

The package also contains an updated version of the enqueue test tool enqt which, however, is only needed for SAP onsite support. Install the parts in the executable directory (SYS/exe/run).

4.1.2 Configuring the enqueue instance

The new enqueue architecture is activated by changing a few profile parameters in the DEFAULT.PFL profile; these parameters are described in “SAP profile parameters” on page 68.

1. Create an instance profile that is used by all components that belong to the enqueue instance. The enqueue instance has its own instance number, instance name, and instance directory.

We chose instance number 00 and instance name EM00. The profile RED_EM00 is shown in Example 4-1:

Example 4-1 SAP profile for enqueue instance

Profile for enqueue server/message server/gateway/syslog collector...

```
SAPSYSTEMNAME = RED
INSTANCE_NAME = EM00
SAPSYSTEM = 00

enqueue/process_location = LOCAL
enqueue/server/replication = true
enqueue/server/threadcount = 3
enqueue/encni/repl_port = 6000
enqueue/backup_file = $(DIR_GLOBAL)/ENQBCK

ipc/shm_psize_16 = 0
ipc/shm_psize_26 = 0
ipc/shm_psize_66 = 0
```

2. Create the instance directory and its subdirectories. In our case, the commands are:

```
mkdir /usr/sap/RED/EM00
mkdir /usr/sap/RED/EM00/data
mkdir /usr/sap/RED/EM00/log
mkdir /usr/sap/RED/EM00/work
```

3. Modify DEFAULT.PFL profile. Example 4-2 shows the entries that need to be changed.

Example 4-2 SAP common profile DEFAULT.PFL

```
SAPDBHOST = $(dbs/db2/hosttcp)
rdisp/mshost = sapred
rdisp/sna_gateway = sapred
rdisp/sna_gw_service = sapgw00
rdisp/vbname = $(rdisp/myname)
# rdisp/enqname
rdisp/btcname = $(rdisp/myname)
enqueue/process_location = REMOTESA
enqueue/serverhost = sapred
enqueue/serverinst = 00
```

4. Add the following parameter to all instance profiles:

```
enqueue/con_retries = 120
```

5. Ensure that the port names used for the enqueue instance are defined in /etc/services on all application servers. (Otherwise, the application server instances will not be able to connect to the enqueue server or the gateway.)

Assuming that the instance number of the enqueue instance is 00, then the following entries are needed:

```
sapdp00      3200/tcp
sapgw00      3300/tcp
```

Starting and stopping of the enqueue instance is described in “Configuring SAP for System Automation” on page 71.

Tip: Save the old DEFAULT.PFL (for example as DEFAULT.CentrallInstance). This will allow you to fall back to the old architecture easily.

To fall back to the central instance architecture, do the following:

- ▶ Stop all SAP instances including the enqueue instance.
- ▶ Restore DEFAULT.CentrallInstance as DEFAULT.PFL.
- ▶ Start the central instance (and optionally, the dialog instances).

All other changes do not affect the ability to start SAP in the old way.

4.1.3 SAP profile parameters

Table 4-1 lists and describes the profile parameters that are related to the enqueue instance.

Table 4-1 SAP profile parameters relevant for the high availability solution

Parameter	Description	Default value	Recommended value
enqueue/serverhost	Host name of the enqueue server.		<virtual hostname>
enqueue/serverinst	Instance number of the enqueue server.		<instance number>
enqueue/process_location	Specifies where the enqueue requests are processed.	OPTIMIZE	REMOTESA (for application servers) LOCAL (for the enqueue server)
enqueue/server/replication	Enables replication.	false	true

Parameter	Description	Default value	Recommended value
enqueue/encni/repl_port	Port number of the enqueue server opens for replication. (The default value is in conflict with the gateway port. Therefore, you must choose a different port if the gateway is part of the enqueue instance.)	3300 + <instance number>	<port number>
enqueue/server/threadcount	Number of I/O threads in the enqueue server.	1	2 or 3
enqueue/backup_file	Specifies where the enqueue server saves the locks on shutdown.	\$(DIR_LOGGING)/ENQBCK	\$(DIR_GLOBAL)/ENQBCK
enqueue/con_retries	Number of seconds the application server tries to reconnect to the enqueue server before an error is indicated to the application.	6	120
rdisp/mshost	Location of the message server.		<virtual hostname>
rdisp/snagateway	Location of the gateway supporting SNA protocol.		<virtual hostname>
rdisp/sna_gw_service	Port name used by the gateway.		sapgw<##>
rslg/collect_daemon/host	Location of the syslog collector.		<virtual hostname>
rdisp/enqname	Application server instance running the (old style) enqueue work process (obsolete).		# comment out
rdisp/btcname	Application server that does the event processing for the batch scheduler (obsolete; every instance can process its own events).		\$(rdisp/myname)

Parameter	Description	Default value	Recommended value
enqueue/encni/repl_port	Port number of the enqueue server opens for replication. (The default value is in conflict with the gateway port. Therefore, you must choose a different port if the gateway is part of the enqueue instance.)	3300 + <instance number>	<port number>
enqueue/server/threadcount	Number of I/O threads in the enqueue server.	1	2 or 3
enqueue/backup_file	Specifies where the enqueue server saves the locks on shutdown.	\$(DIR_LOGGING)/ENQBCK	\$(DIR_GLOBAL)/ENQBCK
enqueue/con_retries	Number of seconds the application server tries to reconnect to the enqueue server before an error is indicated to the application.	6	120
rdisp/mshost	Location of the message server.		<virtual hostname>
rdisp/snagateway	Location of the gateway supporting SNA protocol.		<virtual hostname>
rdisp/sna_gw_service	Port name used by the gateway.		sapgw<##>
rslg/collect_daemon/host	Location of the syslog collector.		<virtual hostname>
rdisp/enqname	Application server instance running the (old style) enqueue work process (obsolete).		# comment out
rdisp/btcname	Application server that does the event processing for the batch scheduler (obsolete; every instance can process its own events).		\$(rdisp/myname)

Parameter	Description	Default value	Recommended value
rdisp/vbname	Application server that runs update work processes (obsolete; update requests are dispatched among appropriate instances automatically).		\$(rdisp/myname)
SAPDBHOST	Location of the database server. (Forward reference to dbs/db2/hosttcp, which is defined in each instance profile to specify the primary database server of the instance while SAPDBHOST is usually defined in DEFAULT.PFL.)		\$(dbs/db2/hosttcp)
ipc/shm_psize_16 ipc/shm_psize_26 ipc/shm_psize_66	Shared memory segments used by enqueue server and replication server. When size is set to 0 the segments are allocated directly, not as pools.		0

4.2 Configuring SAP for System Automation

This section describes startup, monitoring and shutdown procedures that enable System Automation to manage SAP. These scripts are additions to the standard scripts installed by the SAP installation utility. The standard SAP scripts are not touched.

The scripts also write messages to the system console, thereby triggering immediate System Automation actions.

For a comprehensive list of scripts and SAP profiles see Appendix E, “SAP customization for high availability” on page 269.

C-shell and logon profiles

The UNIX applications are invoked by starting the user's default shell and naming the shell script that is to be executed (for example: `/bin/tcsh -c '<command>'`). The C-shell is usually defined as the default shell for the SAP administrator ID.

The C-shell knows four profiles:

- ▶ /etc/csh.cshrc
- ▶ /etc/csh.login
- ▶ \$HOME/.cshrc
- ▶ \$HOME/.login

When the -c option is used, the files /etc/csh.login and \$HOME/.login are *not* processed. This is the case when programs are invoked via BPXBATCH in a started task, or via the System Automation command `INGUSS`. Therefore, make sure that all relevant settings needed for the startup of the SAP system are in the profiles /etc/csh.cshrc and \$HOME/.cshrc.

4.2.1 ICLI servers

The ICLI servers can be started by using a shell script (`iclistart`), or by submitting a job, or by invoking a started task. We decided to use started tasks. For each ICLI server, we created a separate procedure.

If you choose to use the `iclistart` shell script to start the ICLI server, take out the `nohup` and the `&` and add the console message as the last line.

```
export ICLI_TRACE_LEVEL=0
export NLSPATH=/usr/lib/nls/msg/%L/%N
export STEPLIB='DB7X7.SDSNEXIT:DB7X7.SDSNLOAD'
/usr/sbin/fome46ds -PLAN FOME46D -LOGDIR /usr/sap/RED/icli/icli6 -PORT 5006
echo "$_BPX_JOBNAME ENDED" > /dev/console
```

We created `/usr/sap/RED/icli` as a working directory for the ICLI servers. Because all ICLI log files have the process ID in their name, the file names are unique in a sysplex. However, it makes it easier to find the message files of a particular ICLI server if they are written to different directories. Therefore, we created a separate log directory for each ICLI server.

Started tasks or UNIX shell scripts:

Most customers use started tasks to start the ICLI servers. The ICLI server is enabled for operator control and can be stopped using the STOP operator command. System Automation can use the standard MVS mechanism to monitor the started task.

The SAP components are typical UNIX applications. They start off a hierarchy of child processes, and restart some of them dynamically. In some cases, the startup routine ends while the child processes continue to run. Stopping is done by sending UNIX signals to individual processes.

Furthermore, the dependencies and the sequence of starting, stopping and monitoring for the enqueue instance and the application server are complex and cannot be mapped to simple started tasks.

The USS support in System Automation is able to keep track and find the right process, its UNIX process ID, its job name and address space ID. For example, a stop request can be performed by sending a SIGINT signal to the UNIX process first. If it does not stop, a SIGKILL is sent after a while. If this does not help, a CANCEL command on the job name/address space is finally issued. Therefore, for SAP components, it is more appropriate to use the USS support of System Automation and invoke UNIX shell scripts.

4.2.2 Enqueue instance

The enqueue instance is a collection of single-instance SAP resources. They all share the same instance profile and the same instance directory. Namely, they are:

- ▶ Enqueue server
- ▶ Message server
- ▶ Gateway server
- ▶ Syslog collector
- ▶ Syslog sender

And, preferably on a different system:

- ▶ Enqueue replication server

To allow detailed monitoring and faster recovery, all resources are started, stopped and monitored individually. For this purpose, we created the shell script shown in Example 4-3 on page 74.

Example 4-3 Shell script startsap_em00

```
#!/bin/sh
DIR_INSTANCE=/usr/sap/RED/EM00
DIR_EXECUTABLE=/usr/sap/RED/SYS/exe/run
PROFILE=/usr/sap/RED/SYS/profile/RED_EM00

_ES=es.sapRED_EM00
_MS=ms.sapRED_EM00
_CO=co.sapRED_EM00
_SE=se.sapRED_EM00
_GW=gw.sapRED_EM00
_ERS=ers.sapRED_EM00

cd $DIR_INSTANCE/work

case "$1" in

    MS) rm -f $_MS
        ln -s -f $DIR_EXECUTABLE/msg_server $_MS
        $_MS pf=$PROFILE
        ;;

    ES) rm -f $_ES
        ln -s -f $DIR_EXECUTABLE/enserver $_ES
        $_ES pf=$PROFILE
        ;;

    ERS) rm -f $_ERS
        ln -s -f $DIR_EXECUTABLE/enrepserver $_ERS
        $_ERS pf=$PROFILE
        ;;

    CO) rm -f $_CO
        ln -s -f $DIR_EXECUTABLE/rs1gcol1 $_CO
        $_CO -F pf=$PROFILE
        if [ "$?" -gt 0 ]
        then echo "$_BPX_JOBNAME COLLECTOR NOT STARTABLE" > /dev/console
            exit 8
        fi
        ;;

    SE) rm -f $_SE
        ln -s -f $DIR_EXECUTABLE/rs1gsend $_SE
        $_SE -F pf=$PROFILE
        ;;
```

```

GW) rm -f $_GW
ln -s -f $DIR_EXECUTABLE/gwrd $_GW
$_GW pf=$PROFILE
;;

CHECK) $DIR_EXECUTABLE/ensmon pf=$PROFILE 1
if [ "$?" -gt 0 ]
then echo "$_BPX_JOBNAME MONITORING FAILED" > /dev/console
fi
exit $?
;;

*) echo "Missing or wrong parameter $1"
echo "Usage: $0 {MS|ES|ERS|CO|SE|GW|CHECK}"
exit 16

esac
echo "$_BPX_JOBNAME ENDED" > /dev/console

```

The variables `DIR_INSTANCE`, `DIR_EXECUTABLE` and `PROFILE` in the shell script must be adapted to your environment.

The individual components are started as follows:

startsap_em00 ES	Starts the enqueue server
startsap_em00 ERS	Starts the enqueue replication server
startsap_em00 MS	Starts the message server
startsap_em00 GW	Starts the gateway
startsap_em00 CO	Starts the syslog collector
startsap_em00 SE	Starts the syslog sender
startsap_em00 CHECK	Performs a health check on the enqueue server

Important: The SAP license check is based on the CPC node descriptor of the CEC the message server runs on. The CPC node descriptor is displayed with z/OS operator command:

```
D M=CPU
```

The CPC node descriptor is identical for all LPARs on the same CEC. However, if the LPARs are on different CECs, you need to request and install an SAP license key for each CEC. There is technically no limit on the number of license keys you can install.

Run the following command in all LPARs where the message server will run:

```
saplicense -get
```

This will provide you with all hardware keys needed to request the SAP license keys for that SAP system.

4.2.3 Application server instances

We created three shell scripts to start, stop and check local and remote application server instances:

startappsrv <hostname> <##>	Starts an application server instance
stopappsrv <hostname> <##>	Stops an application server instance
checkappsrv <hostname> <##>	Starts an application server monitor

These shell scripts are provided in Appendix E.1, “Shell scripts used for automation” on page 269. The host name (<hostname>) and the instance number (<##>) identify the instance to be managed. This approach is used both for local z/OS application servers and for remote application servers.

What the shell scripts are doing

In the following section, we describe the tasks these shell scripts are involved in.

startappsrv

- First it checks whether the instance is already running by using the SAP utility rfcping (see “rfcping” on page 77). If the instance is running, the shell script indicates the status by sending a message to the system console, and then ends.

This step preserves a running application server instance against unnecessary restarts. For example, in case of an intermittent communication error, checkappsrv terminates and System Automation simply issues the

startappsrv command again. Based on the notification of the active state, System Automation now starts checkappsrv again.

With this approach, System Automation only has to monitor a single process, namely the one started by checkappsrv. The same approach is applicable for both local and remote application servers.

- The application server is started by invoking the following scripts or commands:

```
cleanipc <##> remove  
stopsap_<hostname>_<##>  
startsap_<hostname>_<##>
```

The **cleanipc** and **stopsap** commands ensure that orphan processes or resources are cleaned up before a new **startsap** is performed. If the instance was shut down normally, the **cleanipc** and **stopsap** commands do nothing and end immediately.

If the <hostname> matches the local host name, the commands are executed directly. Otherwise, a remote execution is performed; see “Remote execution” on page 78.

- Finally, it checks periodically until the application server instance is up and responding to rfcping. The successful startup is then indicated by sending a message to the system console.

stopappsrv

- The application server is stopped by invoking the following scripts:

```
stopsap_<hostname>_<##>
```

If the <hostname> matches the local host name, the command is executed directly. Otherwise, a remote execution is performed. See “Remote execution” on page 78.

checkappsrv

- The health check is done by establishing an RFC connection to the application server and periodically checking that it is still responding; see “rfcping” on page 77.

A failure of rfcping indicates that there is (or was) a problem with that instance. Therefore, the existence of this process is used by System Automation to determine the status of the application server instance.

rfcping

This utility is part of the SAP 6.20 kernel and can be downloaded as binary patch from the SAP Service Marketplace, section SAP WEB AS 6.20. The version is compatible with previous SAP releases.

rfcping establishes a RFC connection and retrieves the SAP system information. The command line parameters allow you to choose between different modes.

- ▶ The default option is that rfcping closes the RFC connection and ends after it gets a response from the application server. This is used in the startappsrv script to check whether an application server instance is up and running.
- ▶ Another option specifies that rfcping stays connected and sends a dummy request every few seconds. It only ends if a problem occurs. This mode is used in the checkappsrv script to monitor an application server instance.

We stored the rfcping executable in directory /usr/sap/RED/rfc.

Remote execution

For remote execution, the **rexec** command can be used. This implies that the user ID and password of the remote system is stored in plain text on z/OS. Furthermore, if the password is changed on the remote system, the file must be changed as well.

A better alternative is to use the OpenSSH. This is a secure shell which allows different methods of authentication. It is available as Open Source product on most platforms including z/OS, Linux on zSeries, AIX and Windows.

For more detailed information, refer to the following Web site:

<http://www.openssh.org/>

For Windows, several versions are available; you can refer to the following example:

<http://www.networksimplicity.com/openssh/>

As you can see in the startappsrv script, the remote execution command is executed in background. The reason for this is because **rexec** waits until all started processes have ended or have detached as demons redirecting the standard file descriptors (stdin, stdout, stderr). However, the startsap script invokes saposcol as a normal child process, which implies that the remote execution command waits for saposcol to finish.

See 4.2.3, “Application server instances” on page 76 for a discussion on remote control of application servers by System Automation.

4.2.4 saposcol

The SAP utility saposcol can be started and stopped directly by System Automation. There is no need for a shell script.

You may remove the invocation of saposcol that is done in the standard SAP start scripts, and instead leave the starting and stopping of saposcol solely to System Automation. In startsap_<hostname>_<##> shell scripts for the application server instance(s) that are *running on z/OS*, comment out the following line:

```
start_saposcol;
```

By the way, on z/OS the saposcol messages: WARNING: Effective userid not root! Expect problems! can be ignored because access to the performance data is controlled through a RACF profile.

4.2.5 rfcoscol

The SAP utility rfcoscol is started with the shell script shown in Example 4-4.

Example 4-4 Shell script start_rfcoscol

```
#!/bin/sh
export RFCOSCOL_RETRY=1
export SAP_CODEPAGE=1100      # default
cd /usr/sap/RED/rfc
$DIR_LIBRARY/rfcoscol -DRED_`hostname` -s`
echo "$_BPX_JOBNAME ENDED" > /dev/console
```

The corresponding RFC definition file is located, in our case, in /usr/sap/RED/rfc; Example 4-5 shows the entries.

Example 4-5 RFC definition file saprfc.ini

```
DEST=RED_wtsc42a
TYPE=R
PROGID=wtsc42a.rfcoscol
GWHOST=sapred
GWSERV=sapgw00
RFC_TRACE=0
#
DEST=RED_wtsc04a
TYPE=R
PROGID=wtsc04a.rfcoscol
GWHOST=sapred
GWSERV=sapgw00
RFC_TRACE=0
```

The rfcoscol registers as <hostname>.rfcoscol at the standalone gateway that belongs to the enqueue instance. By using this gateway and the corresponding virtual host name, you ensure that rfcoscol is able to reach the gateway whenever the SAP system is up.

Option RFCOSCOL_RETRY=1 switches on a retry mechanism in case the gateway is currently not running, and rfcoscol keeps trying for a maximum of 24 hours.

If you intend to run more than one rfcoscol instance on the same z/OS system under the same user ID, you need to start them with different process names to allow individual monitoring by System Automation. This can be accomplished by creating a symbolic link and changing the invocation of rfcoscol accordingly. In this case, you must add command line parameter -RFC; see the following example:

```
ln -sf $DIR_LIBRARY/rfcoscol rfcoscol_DEST1
./rfcoscol_DEST1 -RFC -DDEST1_`hostname -s`
```

Also, make sure that the rfcoscols are started with different destinations (DEST entries in the saprfc.ini file), and register at the gateway with a unique PROGID.

Additional SAP setup for RFC connections

Because the standalone gateway server that is started as part of the enqueue instance is guaranteed to be up and reachable whenever that SAP system is up, we propose that RFC servers like RFCOSCOL connect to this gateway.

To reach such an RFC server, this connection must be defined to the SAP system. Using SAP transaction SM59, click **Gateway** and specify the virtual host name and the port name (in our case, sapred and sapgw00); refer to Figure 4-1 on page 81. This must be done for each RFC server that connects to the standalone gateway server.

In SAP transaction AL15, you define the SAPOSCOL destinations. Later on, these can be selected in the CCMS transaction OS07.

You do not have to make the definitions for the RFC connections immediately; you may delay it until the system setup is complete.

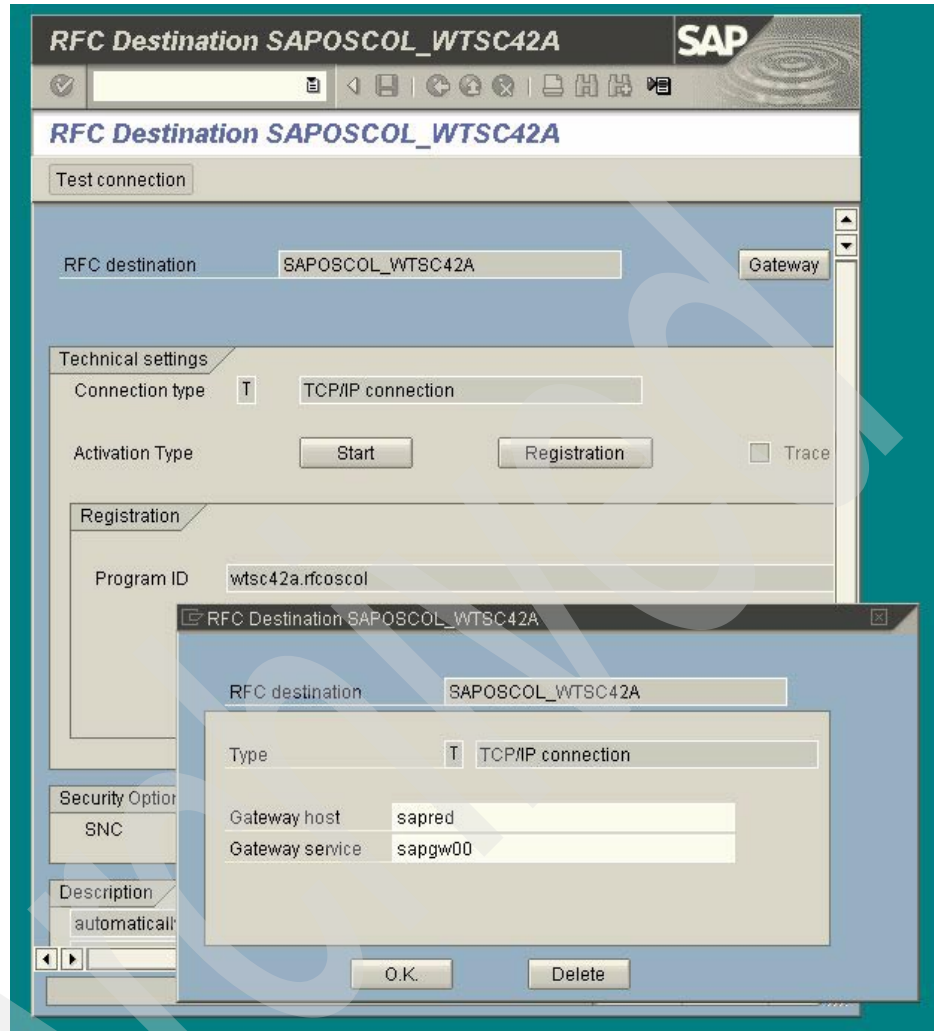


Figure 4-1 Defining the gateway host for rfcoscol with transaction SM59

4.2.6 saprouter

The saprouter can be started and stopped directly by System Automation. There is no need for a shell script.

4.2.7 Summary of start, stop and monitoring commands

Table 4-2 summarizes the start, stop and monitoring commands that are needed when you set up the System Automation policies for SAP.

Table 4-2 Summary of start/stop/monitoring commands

Actions	Value or command
SAP system name User ID Home directory	RED redadm /u/redadm
ICLI servers: - start command (started task) - stop command	S REDICLI# (# = 6, 7, 8, 9) F REDICLI#,APPL=STOP TIMEOUT(60)
VIPA for enqueue instance - start command (started task)	S TCPVIPA,VIPA=172.20.10.1
Enqueue server: - start command - stop command - process name to be monitored - additional monitor	/u/redadm/startsap_em00 ES /bin/kill -2 %PID% ./es.sapRED_EM00 /u/redadm/startsap_em00 CHECK
Enqueue replication server: - start command - stop command - process name to be monitored	/u/redadm/startsap_em00 ERS /bin/kill -2 %PID% ./ers.sapRED_EM00
Message server: - start command - stop command - process name to be monitored	/u/redadm/startsap_em00 MS /bin/kill -2 %PID% ./ms.sapRED_EM00
Gateway: - start command - stop command - process name to be monitored	/u/redadm/startsap_em00 GW /bin/kill -2 %PID% ./gw.sapRED_EM00
Syslog collector: - start command - stop command - process name to be monitored	/u/redadm/startsap_em00 CO /bin/kill -2 %PID% ./co.sapRED_EM00
Syslog sender: - start command - stop command - process name to be monitored	/u/redadm/startsap_em00 SE /bin/kill -2 %PID% ./se.sapRED_EM00

Actions	Value or command
Application server instances: <ul style="list-style-type: none"> - start command - poststart (monitor) command - stop command - process name to be monitored 	/u/redadm/startappsrv <hostname> <##> /u/redadm/checkappsrv <hostname> <##> /u/redadm/stoppappsrv <hostname> <##> /usr/sap/RED/rfc/rfcping_<hostname>_<##>
saposcol: <ul style="list-style-type: none"> - start command - stop command - process name to be monitored 	/usr/sap/RED/SYS/exe/run/saposcol -l /bin/kill -2 %PID% /usr/sap/RED/SYS/exe/run/saposcol
rfcoscol: <ul style="list-style-type: none"> - start command - stop command - process name to be monitored 	/u/redadm/start_rfcosc /bin/kill -2 %PID% /usr/sap/RED/SYS/exe/run/rfcosc
VIPA for saprouter: <ul style="list-style-type: none"> - start command (started task) 	S TCPVIP,VIPA=172.20.10.3
saprouter: <ul style="list-style-type: none"> - start command - stop command - process name to be monitored 	/usr/sap/RED/SYS/exe/run/saprouter -r /usr/sap/RED/SYS/exe/run/saprouter -s /usr/sap/RED/SYS/exe/run/saprouter

Customizing System Automation for OS/390 for SAP

This chapter shows you how to set up System Automation for OS/390 (SA OS/390) for SAP High Availability.

Note that, along with these installation instructions, detailed knowledge of SA OS/390 is required to make SAP HA work.

5.1 Preparing SA OS/390 for SAP HA

In this section, we describe what you need to do before you define the SAP-related components in the SA OS/390 policy.

5.1.1 Before you start

If you have not already done so, refer to 3.7, “Netview and System Automation basic setup” on page 54. Verify the following:

- ▶ NetView is customized and running.
- ▶ SA OS/390 is customized and running.
- ▶ Automated Restart Manager (ARM) does not interfere with SA OS/390.
- ▶ Either the NetView Management Console (NMC) or the Status Display Facility (SDF) is customized and working.
- ▶ You can stop and start the systems using SA OS/390.

5.1.2 Setting initialization defaults for SA OS/390 (AOFEXDEF)

Add the following variables to the default initialization exit AOFEXDEF and concatenate the two variables to the GLOBALV PUTC command:

- ▶ AOFRESTARTALWAYS = 0

With this parameter, SA OS/390 will not to restart a resource that has been shut down outside its control, if that resource has reached its critical error threshold.

This is necessary, for example, for the NFS server. If the NFS server encounters an internal error, it stops gracefully. Without this option, SA OS/390 will try to restart it forever on the same system.

- ▶ AOFUSSWAIT = 30

AOFUSSWAIT is the time SA OS/390 waits for the completion of a user-specified z/OS UNIX monitoring routine (defined in the z/OS UNIX Control Specification panel) until it gets a timeout. When the timeout occurs, SA OS/390 no longer waits for the response from the monitoring routine and sends a SIGKILL to that routine.

For SAP HA, we increase the value from 10 seconds (default) to 30 seconds, mainly because we run many monitoring routines and we want to decrease the amount of messages to the NetView netlog and syslog.

For more details, refer to Appendix B, “Global Variables to Enable Advanced Automation”, in *System Automation for OS/390: Customizing and Programming*,

SC33-7035, and to the white paper *System Automation for OS/390: Enhancements for OS/390 UNIX System Services Automation*. This white paper can be downloaded from the System Automation for OS/390 Web site at the following URL:

<http://www.ibm.com/servers/eserver/zseries/software/sa/sainfos.html>

5.1.3 Setting the region size for NetView to 2 GB

Set the region size of the NetView started procedure to 2 GB (or 0, which gives you the maximum storage you can get), as shown in Example 5-1:

Example 5-1 Setting the region size for NetView to 2 GB

```
//HSAAPPL PROC PROG=DSIMNT,      ** PGM USED TO START NETVIEW
//                               ** REGION SIZE(IN M) FOR NETVIEW
```

If the region size of the NetView started procedure is too small, you may receive the error message shown in Figure 5-1:

```
EA995I SYMPTOM DUMP OUTPUT
  USER COMPLETION CODE=4091 REASON CODE=0000000C
  TIME=14.34.23 SEQ=05730 CPU=0000 ASID=00D1
  PSW AT TIME OF ERROR 078D1000 89E3555A ILC 2 INTC 0D
  NO ACTIVE MODULE FOUND
  NAME=UNKNOWN
  DATA AT PSW 09E35554 - 00181610 0A0D47F0 B10A1811
  AR/GR 0: 153B8498/84000000 1: 00000000/84000FFB
        2: 00000000/0000000C 3: 00000000/00000001
        4: 00000000/09ADCC60 5: 00000000/14BA67D8
        6: 00000000/14BB3B48 7: 00000000/14BB3FB8
        8: 00000000/00FCB210 9: 00000000/00000030
        A: 00000000/00000004 B: 00000000/89E35488
        C: 00000000/14BB50F8 D: 00000000/153B87F0
        E: 14BB3FB8/00000000 F: 14BB3B48/0000000C
  END OF SYMPTOM DUMP
  BPXP009I THREAD 12BA416000000001, IN PROCESS 84412019, ENDED
  ABNORMALLY WITH COMPLETION CODE 84000FFB, REASON CODE 0000000C.
```

Figure 5-1 Error message when the region for NetView is too small

5.1.4 Customizing Status Display Facility (SDF)

The Status Display Facility (SDF) is used to monitor system resources on the local z/OS system, as well as on other systems. The resources are monitored by noting the colors in which they appear, each color representing a different state.

The drawback of the standard SDF screens is that you can only monitor the status of resources of one system at a time. In our case, we developed a customized SDF panel, which combines on one screen the status of all SAP-related resources running on all LPARs. This is very helpful, for example, to see applications moving between LPARs.

Figure 5-2 is a screenshot of our SDF panel AOFSAP.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1	SAP_RTVIPA	SAP_RTVIPA
RED_DB2IRLM	RED_DB2IRLM	SAP_ROUTER	SAP_ROUTER
RED_DB2DIST	RED_DB2DIST	RED_VIPA	RED_VIPA
RED_DB2SPAS	RED_DB2SPAS	RED_ES	RED_ES
RED_RFC	RED_RFC	RED_MS	RED_MS
REDICLI6	REDICLI6	RED_GW	RED_GW
REDICLI7	REDICLI7	RED_CO	RED_CO
REDICLI8	REDICLI8	RED_SE	RED_SE
REDICLI9	REDICLI9	RED_ERS	RED_ERS
APPSRV11	APPSRV10	APPSRV06	APPSRV06
SAP_QSCOL	SAP_QSCOL	APPSRV07	APPSRV07
		APPSRV08	APPSRV08

06/06/02 13:21

PF1=HELP 2=DETAIL 3=END 6=ROLL 7=UP 8=DN 9=DEL 10=LF 11=RT 12=TOP

MA a 23/00

Figure 5-2 Example of the SDF screen AOFSAP

Our definitions, including the new SDF panel AOFSAP, the modified SDF tree definition member AOFTSC04, and the modified SDF start screen AOFPSYST, can be found in Appendix D.1, “Status Display Facility definition” on page 258. These samples can be used as a base to build your own customized SDF panel.

A detailed description of how to customize SDF can be found in *System Automation for OS/390: Programmer's Reference*, SC33-7043. Of course, you can also use the NetView Management Console (NMC) to monitor SAP application status.

5.1.5 Sending UNIX messages to the syslog

Add the following entry to the syslog configuration file `/etc/syslog.conf` to send UNIX syslogd messages to the z/OS syslog:

```
*.* /dev/console
```

UNIX messages will appear in the z/OS syslog with a BPXF024I message id.

5.1.6 Setting MAXFILEPROC in BPXPRMxx

The USS parameter MAXFILEPROC, which is defined in the member BPXPRMxx of the PARMLIB, should be set to a “reasonable” value. It must not be set to the maximum of 65,536.

This parameter influences the size of the file table that is allocated in each UNIX process. If the value is too high, SA OS/390 will not be able to issue multiple INGUSS commands in parallel; the INGUSS commands will fail with an error message saying that a resource is temporarily not available. *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966 recommends that you set it to 2,000.

5.2 Defining the SAP-related resources

In this section, we describe the implementation of the applications and groups that we defined in our SA OS/390 policy.

Note: We provide you with our SA OS/390 policy database; for information on how to retrieve it, refer to Appendix F, “Additional material” on page 279.

5.2.1 Overview of the resources

The following SAP-related components must be defined in the SA OS/390 policy:

- Resources that are related to a specific SAP system (in our case, RED):
 - Database server
 - Enqueue instance, including enqueue server, message server, gateway, syslog collector, and syslog sender
 - Enqueue replication server
 - Application servers (both local and remote)
 - Local applications: ICL servers and rfcsccl

- ▶ Resources that are common to all the SAP systems:
 - NFS server
 - saprouter
 - Local applications: saposcol

5.2.2 Naming convention

Table 5-1 summarizes the naming convention we used.

Table 5-1 Naming convention

Type of resource	Naming convention
Resources related to SAP system RED	RED_*
Resources related to SAP in general	SAP_*
Groups with system scope	*GRP
Groups with sysplex scope	*PLEX
Jobnames for SAP RED	REDADM*
Jobnames for general SAP	SAP*

5.2.3 Classes

A *class* represents a policy that is common to one or more applications. It can be used as a template to create new applications.

In our environment, we used three classes:

- ▶ The default UNIX System Services class: USS_APPLICATION
- ▶ One class for the DB2 MSTR address space: CLASS_DB2_MSTR
- ▶ One class for the other DB2 address spaces: CLASS_RED_DB2_CHILDS

USS_APPLICATION

This class is provided with the sample policy database of SA OS/390. All UNIX resources must refer to this class.

Attention: Any abnormal end of a UNIX application will appear to SA OS/390 as a *shutdown outside of automation* condition. Since we want SA OS/390 to recover from these situations, we must change the restart option to ALWAYS.

Example 5-2 on page 91 shows the definition of this class.

Example 5-2 Definition of class USS_APPLICATION

Entry Name: USS_APPLICATION

Object Type: CLASS

Automation Info

Start Timeout. . . . 00:00:30
Monitor Routine. . . AOFUXMON
Periodic Interval. . 00:10
Restart Option . . . ALWAYS
Shut Delay 00:00:30
Term Delay 00:00:02

CLASS_DB2_MASTER

This class is used for defining the DB2 master address space for all DB2 subsystems running in the sysplex. Example 5-3 shows the definition of this class.

Example 5-3 Definition of class CLASS_DB2_MASTER

Entry Name: CLASS_DB2_MASTER

Object Type: CLASS

Relationships

HASPARENT JES2/APL/=

Startup

MAINT
MVS &SUBSCMDPFX STA DB2 ACCESS(MAINT) &EHKVAR1
NORM
MVS &SUBSCMDPFX STA DB2 &EHKVAR1

Shutdown NORM

1
INGRDTTH &SUBSAPPL S

Shutdown IMMED

1
MVS &SUBSCMDPFX STOP DB2,MODE(FORCE)
2
MVS C &SUBSJOB

Shutdown FORCE

1
MVS &SUBSCMDPFX STOP DB2,MODE(FORCE)
2
MVS C &SUBSJOB

CLASS_RED_DB2_CHILDS

This class is used for defining the subordinate DB2 address spaces (DBM1, DIST, IRLM and SPAS) for the DB2 subsystem related to SAP RED.

The subordinate resources are defined for monitoring purposes only. Therefore, they are defined with the attributes “External startup” and “External shutdown” set to ALWAYS. Example 5-4 shows the definition of this class.

Example 5-4 Definition of class CLASS_RED_DB2_CHILDS

Entry Name: CLASS_RED_DB2_CHILDS

Object Type: CLASS

Automation Info

External Startup . ALWAYS

External Shutdown. ALWAYS

Relationships

HASPARENT . . . RED_DB2MSTR/APL/=

Condition . . . StartsMeAndStopsMe

5.2.4 Database server

In this section, we provide the definition of the DB2 subsystem related to SAP RED. It consists of a DB2 data sharing group with two members: D7X1 running on SC42, and D7X2 running on SC04.

System definition

There is one DB2 member running on each LPAR. By cloning the resource definitions, we avoid having to define resources that are alike for every subsystem. The cloning variables are defined as part of the system definition.

Example 5-5 on page 93 shows how the name of the DB2 subsystem D7X2 is defined in the &AOCCLONE2 variable of system SC04.

Example 5-5 Specifying system information

Entry Type : System	PolicyDB Name : SAP_HA_SAP
Entry Name : SC04	Enterprise Name : SAP_HA
Operating system MVS MVS VM TPF VSE CF LINUX	
Specify information (MVS systems only):	
MVS SYSNAME. SC04	MVS system name
Clone Id 04	&AOCCCLONE.
Clone Id 1	&AOCCCLONE1.
Clone Id 2 D7X2	&AOCCCLONE2.

Applications

We define one application per DB2 address space: MSTR, DBM1, DIST, IRLM, and SPAS.

RED_DB2MSTR

This application corresponds to the DB2 MSTR address space. Example 5-6 shows the definition of the application RED_DB2MSTR.

Note: We have chosen to let Automatic Restart Manager (ARM) recover DB2 failures (see 3.5.2, “ARM policy” on page 52). Therefore, we set the critical threshold number to 1 to tell SA OS/390 not to recover the resource. You could also use SA OS/390 for the recovery of the DB2 system.

Example 5-6 Definition of the application RED_DB2MSTR

Entry Name: RED_DB2MSTR							
Link to Class CLASS_DB2_MSTR							
Application Information							
Application Type. . . DB2							
Subtype MSTR							
Clone Job Name. . . . YES							
Job Name. &AOCCCLONE2.MSTR							
Automation Information							
Command Prefix. . . . &AOCCCLONE2.							
Thresholds							
		Critical		Frequent		Infrequent	
Resource	Number	Interval	Number	Interval	Number	Interval	
DB2_MASTER	1	00:30	3	00:30	2	00:30	

RED_DB2DBM1

This application corresponds to the DB2 DBM1 address space. Example 5-7 shows the definition of the application RED_DB2DBM1.

Example 5-7 Definition of the application RED_DB2DBM1

Entry Name: RED_DB2DBM1

Link to Class CLASS_RED_DB2_CHILDS

Application Information

Application Type. . . DB2

Subtype DBM1

Clone Job Name. . . . YES

Job Name. &AOCCCLONE2.DBM1

RED_DB2DIST

This application corresponds to the DB2 DIST address space. Example 5-8 shows the definition of the application RED_DB2DIST.

Example 5-8 Definition of the application RED_DB2DIST

Entry Name: RED_DB2DIST

Link to Class CLASS_RED_DB2_CHILDS

Application Information

Application Type. . . DB2

Subtype DIST

Clone Job Name. . . . YES

Job Name. &AOCCCLONE2.DIST

RED_DB2IRLM

This application corresponds to the DB2 IRLM address space. Example 5-9 shows the definition of the application RED_DB2IRLM.

Example 5-9 Definition of the application RED_DB2IRLM

Entry Name: RED_DB2IRLM

Link to Class CLASS_RED_DB2_CHILDS

Application Information

Application Type. . . DB2

Subtype IRLM

Clone Job Name. . . . YES

Job Name. &AOCCCLONE2.IRLM

RED_DB2SPAS

The application RED_DB2SPAS corresponds to the DB2 SPAS address space. Example 5-10 shows the definition of the application RED_DB2SPAS.

Example 5-10 Definition of the application RED_DB2SPAS

Entry Name: RED_DB2SPAS
Link to Class CLASS_RED_DB2_CHILDS

Application Information

Application Type. . .	DB2
Subtype	SPAS
Clone Job Name. . .	YES
Job Name.	&A0CCLONE2.SPAS

Application groups

After having defined the applications, we group them as shown in Figure 5-3. One DB2 subsystem is to be active on each LPAR (active applications are represented as shaded boxes).

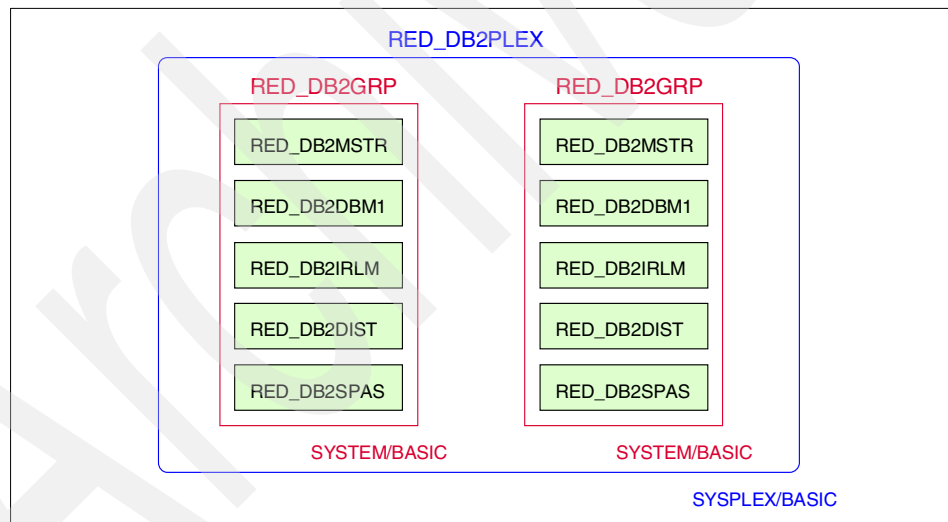


Figure 5-3 RED_DB2PLEX application group

RED_DB2GRP

This SYSTEM group combines the DB2 applications on a single system. Example 5-11 on page 96 shows the definition of the application group RED_DB2GRP.

Example 5-11 Definition of the application group RED_DB2GRP

```
Entry Type: ApplicationGroup
Entry Name: RED_DB2GRP
Application Group Type . SYSTEM
Nature . . . . . BASIC

Select applications:
  RED_DB2DBM1
  RED_DB2DIST
  RED_DB2IRLM
  RED_DB2MSTR
  RED_DB2SPAS

Relationships
  Relationship Type . . HASPARENT
  Supporting Resource . JES2/APL/=
```

RED_DB2PLEX

This superior application group is of scope SYSPLEX. It determines that the application group RED_DB2GRP is to be activated on the two specified systems SC04 and SC42. Example 5-12 shows the definition of the application group RED_DB2PLEX.

Example 5-12 Definition of the application group RED_DB2PLEX

```
Entry Type: ApplicationGroup
Entry Name: RED_DB2PLEX
Application Group Type . SYSPLEX
Nature . . . . . BASIC

Select resources:
  RED_DB2GRP/APG/SC04
  RED_DB2GRP/APG/SC42
```

5.2.5 Enqueue instance and enqueue replication server

In this section, we provide the definition of the enqueue instance. And because it is closely related, we also describe the definition of the enqueue replication server.

Applications

We define one application per component of the enqueue instance: enqueue server, message server, syslog collector, syslog sender, SAP gateway, and VIPA associated with the enqueue instance. Another application is defined for the enqueue replication server.

RED_ES

This application corresponds to the enqueue server. Example 5-13 shows the definition of the application RED_ES.

Example 5-13 Definition of the application RED_ES

```
Entry Name: RED_ES
Link to Class USS_APPLICATION

Application Information
  Application Type. . . USS
  Job Name. . . . . REDADMES

Relationships
  Relationship Type . . MAKEAVAILABLE
  Supporting Resource . RED_VIPA/APL/=
  Automation. . . . . PASSIVE
  Chaining. . . . . WEAK
  Condition . . . . . WhenAvailable

Startup
  INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/startsap_em00 ES >&
  /u/redadm/startsap_em00_ES.&SYSNAME..log'

Shutdown NORM
  1
  INGUSS /bin/kill -2 %PID%
  4
  INGUSS /bin/kill -9 %PID%

Thresholds
  Critical      Frequent      Infrequent
  Resource      Number  Interval  Number  Interval  Number  Interval
  RED_ES        1      01:00      1      02:00      1      12:00

USS Control
  User ID. . . . . REDADM
  Command/Path . . . . ./es.sapRED_EM00
```

Note that the critical threshold number of the enqueue server is set to 1. This means that SA OS/390 will *not* try to restart the enqueue server on the same LPAR. Instead, a failover will be triggered whenever the enqueue server terminates.

Optionally, a second monitor routine can be defined for the enqueue server.

Important: Later tests and early customer experience have shown that the availability of the enqueue server is not enhanced by using the second monitor routine. In fact, false alarms of this monitor cause the opposite effects. Therefore, we recommend that you do not use this second monitor routine to trigger a restart of the enqueue server.

Note that the described poststart command continues to run when the enqueue server is down. It would be more appropriate to use the INGTIMER command to invoke a REXX procedure that then calls the monitor routine. Change or remove this poststart command if you use the sample policy that is distributed with this redbook.

The second monitor is started by the following poststart command:

```
AFTER 05,EVERY 01,ID=CHECKES,INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/
startsap_em00 CHECK >& /u/redadm/startsap_em00_CHECK.&SYSNAME..log'
```

This starts the monitor 5 minutes after the UP status has been reached. Then it runs every minute. If it fails, it writes out the following message:

```
BPXF024I (REDADM) REDADMES MONITORING FAILED
```

This message is trapped in the automation table and the generic routine ISSUECMD is called, as shown in Example 5-14.

Example 5-14 Extract of the SA OS/390 table

```
IF MSGID = 'BPXF024I' & DOMAINID = 'SC04A' THEN BEGIN;
IF TOKEN(4)='MONITORING' & TOKEN(5) = 'FAILED'.
    THEN EXEC(CMD('ISSUECMD AUTOTYP=START'))
    ROUTE(ONE %AOFOPGSSOPER%));
```

Two commands will be issued: the first command restarts the enqueue server, and the second command removes the vote after one minute; see Example 5-15.

Example 5-15 Optional definition for the application RED_ES

Message ID : BPXF024I

Commands:

```
INGREQ RED_ES,REQ=STOP,VERIFY=NO,PRECHECK=NO,OVERRIDE=DPY,
SCOPE=ONLY,RESTART=YES,OUTMODE=LINE
```

```
AFTER 00:01:00,INGSET CANCEL RED_ES,REQUEST=MAKEUNAVAILABLE_ONLY,
SOURCE=AUTOOPS,VERIFY=NO,OUTMODE=LINE
```

RED_MS

This application corresponds to the message server. Example 5-16 shows the definition of the application RED_MS.

Example 5-16 Definition of the application RED_MS

```
Entry Name: RED_MS
Link to Class USS_APPLICATION

Application Information
  Application Type. . . USS
  Job Name. . . . . REDADMMS
  JCL Procedure Name. . USSSTART

Startup
  INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/startsap_em00 MS >&
  /u/redadm/startsap_em00_MS.&SYSNAME..log'

Shutdown NORM
  1
  INGUSS /bin/kill -2 %PID%
  4
  INGUSS /bin/kill -9 %PID%

USS Control
  User ID. . . . . REDADM
  Command/Path . . . . ./ms.sapRED_EM00
```

RED_CO

This application corresponds to the syslog collector. The purpose of the relationship definitions is explained in “RED_COPLEX” on page 107. Example 5-17 shows the definition of the application RED_CO.

Example 5-17 Definition of the application RED_CO

```
Entry Name: RED_CO
Link to Class USS_APPLICATION

Application Information
  Application Type. . . USS
  Job Name. . . . . REDADMCO

Relationships
  Relationship Type . . PREPAVAILABLE
  Supporting Resource . RED_COPLEX/APG
  Automation. . . . . PASSIVE
  Chaining. . . . . WEAK
  Condition . . . . . WhenObservedDown
```

Relationship Type . . MAKEAVAILABLE
Supporting Resource . RED_COMPLEX/APG
Automation. PASSIVE
Chaining. WEAK
Condition WhenObservedDown

Startup

INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/startsap_em00 CO >&
/u/redadm/startsap_em00_CO.&SYSNAME..log'

Shutdown NORM

1
INGUSS /bin/kill -2 %PID%
4
INGUSS /bin/kill -9 %PID%

USS Control

User ID. REDADM
Command/Path/co.sapRED_EM00

RED_SE

This application corresponds to the syslog sender. Example 5-18 shows the definition of the application RED_SE.

Example 5-18 Definition of the application RED_SE

Entry Name: RED_SE

Link to Class USS_APPLICATION

Application Information

Application Type. . . USS
Job Name. REDADMSE

Startup

INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/startsap_em00 SE >&
/u/redadm/startsap_em00_SE.&SYSNAME..log'

Shutdown NORM

1
INGUSS /bin/kill -2 %PID%
4
INGUSS /bin/kill -9 %PID%

USS Control

User ID. REDADM
Command/Path/se.sapRED_EM00

RED_GW

This application corresponds to the SAP gateway. Example 5-19 shows the definition of the application RED_GW.

Example 5-19 Definition of the application RED_GW

Entry Name: RED_GW

Link to Class USS_APPLICATION

Application Information

Application Type. . . USS

Job Name. REDADMGW

Startup

INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/startsap_em00 GW >&
<u/redadm/startsap_em00_GW.&SYSNAME..log'

Shutdown NORM

1

INGUSS /bin/kill -2 %PID%

4

INGUSS /bin/kill -9 %PID%

USS Control

User ID. REDADM

Command/Path/gw.sapRED_EM00

RED_VIPA

This application corresponds to the VIPA associated with the enqueue instance. Example 5-20 shows the definition of the application RED_MS.

Example 5-20 Definition of the application RED_VIPA

Entry Name: RED_VIPA

Application Information

Application Type. . . STANDARD

Job Name. TCPVIPAI

JCL Procedure Name. . TCPVIPA

Application Automation Definition

Job Type. TRANSIENT

Transient Rerun . . . YES

Startup

Parameters. ,VIPA='172.20.10.1'

Messages

ACORESTART

```
INGGROUP RED_ERSPLEX/APG,ACTION=ADJUST,  
    MEMBERS=(RED_ERS/APL/&SYSNAME.),PREF=(1)  
RUNNING  
INGGROUP RED_ERSPLEX/APG,ACTION=RESET  
  
INGGROUP RED_ERSPLEX/APG,ACTION=ADJUST,  
    MEMBERS=(RED_ERS/APL/&SYSNAME.),PREF=(1)
```

RED_ERS

This application corresponds to the enqueue replication server.

Via the relationship definitions with the enqueue instance members, we establish the following dependencies between the enqueue server and the enqueue replication server:

- ▶ The enqueue replication server is always started on a different LPAR from the one on which the replication server is running (1).
- ▶ If the enqueue server fails, it will be attracted by the enqueue replication server and will restart on the LPAR where the enqueue replication server is running (2).
- ▶ The enqueue replication server is not started before the enqueue server is in an observed DOWN status (3).

The INGGROUP commands in the application automation definitions of the RED_VIPA resource (refer to Example 5-20 on page 101) ensure that the enqueue replication server is not started where the enqueue server (actually the related VIPA) is currently running (1).

The INGGROUP commands in the startup poststart definitions of the RED_ERS resource (see Example 5-21 on page 103) ensure that the enqueue replication server attracts the enqueue server if this fails (2).

The MAKEAVAILABLE WhenObservedSoftDown relationship against RED_EMGRP/APG/= will prevent the start of RED_ERS whenever the RED_EMGRP on the same system is in HARDDOWN status (3).

In a two-LPAR environment, this may prevent the enqueue replication server from restarting at all. You may want to set a BROKEN enqueue server to AUTODOWN as soon as it is restarted on the other system, in order to allow the enqueue replication server to restart.

Example 5-21 on page 103 shows the definition of the application RED_ERS.

Example 5-21 Definition of the application RED_ERS

Entry Name: RED_ERS

Link to Class USS_APPLICATION

Application Information

Application Type. . . USS

Job Name. REDADMER

Relationships

Relationship Type. . PREPAVAILABLE

Supporting Resource. RED_ERSPLEX/APG

Automation PASSIVE

Chaining WEAK

Condition WhenObservedDown

Relationship Type. . MAKEAVAILABLE

Supporting Resource. RED_ERSPLEX/APG

Automation PASSIVE

Chaining WEAK

Condition WhenObservedDown

Relationship Type. . MAKEAVAILABLE

Supporting Resource. RED_EMGRP/APG/=

Automation PASSIVE

Chaining WEAK

Condition WhenObservedSoftDown

Relationship Type. . HASPPARENT

Supporting Resource. TCPIP/APL/=

Messages

ACORESTART

INGGROUP RED_EMPLX/APG,ACTION=ADJUST,
MEMBER=(RED_EMGRP/APG/&SYSNAME.),PREF=(700)

Startup STARTUP

INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/startsap_em00 ERS >&
/u/redadm/startsap_em00_ERS.&SYSNAME..log'

Startup POSTSTART

INGGROUP RED_EMPLX/APG,ACTION=RESET

INGGROUP RED_EMPLX/APG,ACTION=ADJUST,
MEMBER=(RED_EMGRP/APG/&SYSNAME.),PREF=(700)

Shutdown NORM

1

INGUSS /bin/kill -2 %PID%

```
4
INGUSS /bin/kill -9 %PID%

USS Control
User ID. . . . . REDADM
Command/Path . . . . ./ers.sapRED_EM00
```

Application groups

First, we define a SYSTEM application group to combine the components of the enqueue instance. Then, we implement two SYSPLEX groups: one for the enqueue instance, the other for the enqueue replication server. Finally, we create a nested SYSPLEX group structure, including a MOVE group for the VIPA, and another MOVE group for the syslog collector.

RED_EMGRP

This SYSTEM group combines the components of the enqueue instance. Example 5-22 shows the definition of the application group RED_EMGRP.

Example 5-22 Definition of the application group RED_EMGRP

```
Entry Type: ApplicationGroup
Entry Name: RED_EMGRP
Application Group Type . SYSTEM
Nature . . . . . BASIC

Select applications:
  RED_CO
  RED_ES
  RED_GW
  RED_MS
  RED_SE
  RED_VIPA

Relationships
  Relationship Type . . HASPPARENT
  Supporting Resource . TCPIP/APL/=
```

RED_EMPLEX and RED_ERSPLEX

Two superior SYSPLEX/MOVE application groups must be defined: one for the enqueue instance (RED_EMPLEX), and the other one for the enqueue replication server (RED_ERSPLEX).

This will ensure that only one enqueue instance and one enqueue replication server are started at a time, and that they are running on different systems, as shown in Figure 5-4 (active applications are represented as shaded boxes).

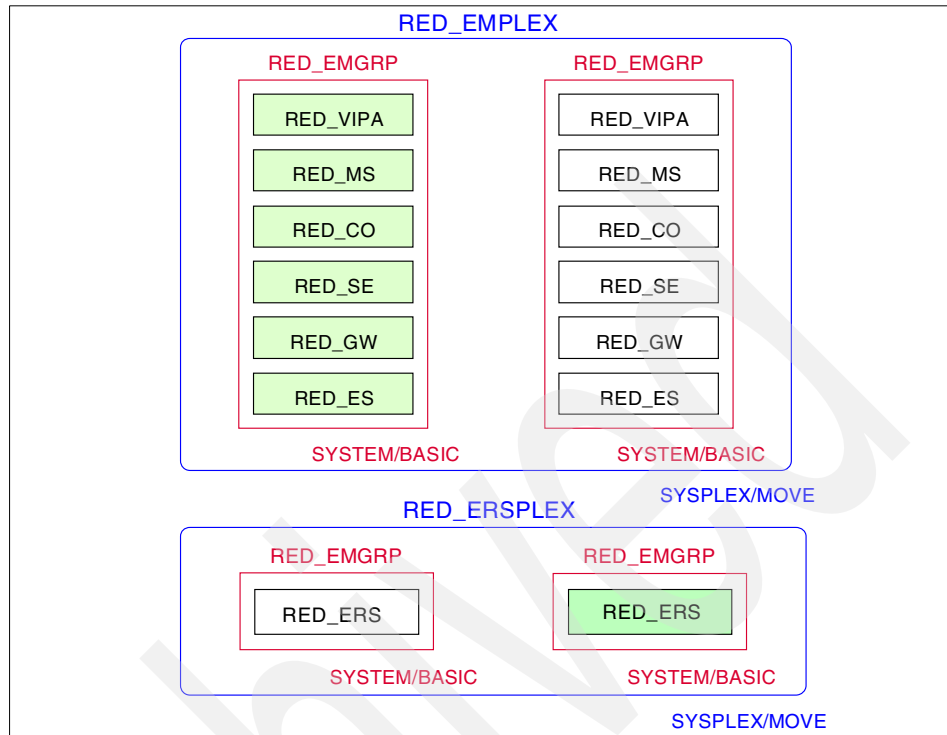


Figure 5-4 RED_EMPLEX and RED_ERSPLEX application groups

Example 5-23 shows the definition of the application group RED_EMPLEX.

Example 5-23 Definition of the application group RED_EMPLEX

```

Entry Type: ApplicationGroup
Entry Name: RED_EMPLEX
Application Group Type . SYSPLEX
Nature . . . . . MOVE
Default Preference . . . 601

```

```

Select resources:
  RED_EMGRP/APG/SC04
  RED_EMGRP/APG/SC42

```

Example 5-24 on page 106 shows the definition of the application group RED_ERSPLEX.

Example 5-24 Definition of the application group RED_ERSPLEX

```
Entry Type: ApplicationGroup
Entry Name: RED_ERSPLEX
Application Group Type . SYSPLEX
Nature . . . . . MOVE
Default Preference . . . 601

Select applications:
    RED_ERS

Relationships
    Relationship Type. . MAKEAVAILABLE
    Supporting Resource. RED_VPLEX/APG
    Automation . . . . . PASSIVE
    Chaining . . . . . WEAK
    Condition . . . . . WhenAvailable

    Relationship Type. . PREPAVAILABLE
    Supporting Resource. RED_VPLEX/APG
    Automation . . . . . PASSIVE
    Chaining . . . . . WEAK
    Condition . . . . . WhenAvailable
```

RED_VPLEX

This application group is a SYSPLEX/MOVE PASSIVE group defined for the VIPA associated with the enqueue instance. Its purpose is to define a relationship between the enqueue server and its VIPA. This ensures that the INGGROUP command in the application automation definitions of the RED_VIPA resource (see Example 5-20 on page 101) is processed by SA OS/390 prior to the decision where to place the enqueue replication server.

Since RED_VIPA is a MOVE group, only one of the applications in the group is started at a time, as shown in Figure 5-5 (active applications are represented as shaded boxes).

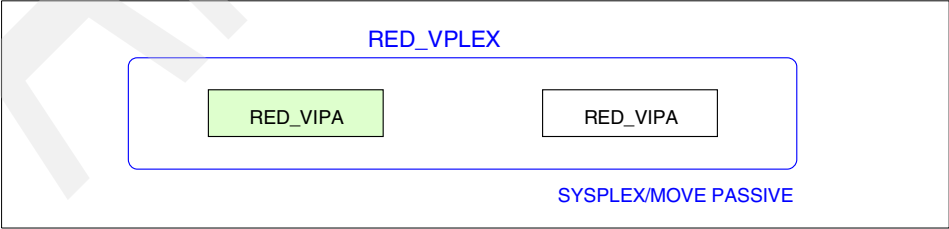


Figure 5-5 RED_VPLEX application group

Example 5-25 shows the definition of the application group RED_VPLEX.

Example 5-25 Definition of the application group RED_VPLEX

Entry Type: ApplicationGroup
Entry Name: RED_VPLEX
Application Group Type . SYSPLEX
Nature MOVE
Behaviour. PASSIVE

Select applications:
RED_VIPA

RED_COPLEX

This application group is a SYSPLEX/MOVE PASSIVE group defined for the syslog collector. Its purpose is to ensure that only one collector daemon is started or active at a time, as shown in Figure 5-6 (active applications are represented as shaded boxes).

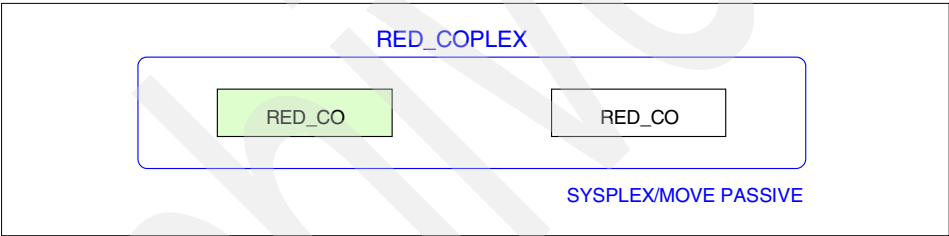


Figure 5-6 RED_COPLEX application group

Example 5-26 shows the definition of the application group RED_COPLEX.

Example 5-26 Definition of the application group RED_COPLEX

Entry Type: ApplicationGroup
Entry Name: RED_COPLEX
Application Group Type . SYSPLEX
Nature MOVE
Behaviour. PASSIVE

Select applications:
RED_CO

5.2.6 Application servers

In this section, we provide the definitions of the application servers (both local and remote).

Applications

We define one application per application server: APPSRV06 running on VMLINUX6, APPSRV10 running on SC42, and APPSRV11 running on SC04.

APPSRV06

This application corresponds to the remote application server running on VMLINUX6.

Because this application server is running on a remote Linux for z/OS system, it can not be “seen” by SA OS/390. When started, the only indication for an up and running status is the response of the monitor routine.

For this remote application server, we defined two STOP commands:

- ▶ One SHUTNORM command, which kills only the monitor routine. When the monitor routine is gone, the remote application server appears to be down for SA OS/390.

After a move of the resource to a different LPAR, the new monitor routine will just “reconnect” to the application server, which is still running. If you want to stop an LPAR and move all applications to another one, the SHUTNORM command is sufficient.

- ▶ One SHUTFORCE command, which really does stop the application server.

Example 5-27 shows the definition of the application APPSRV06.

Example 5-27 Definition of the application APPSRV06

Entry Name: APPSRV06

Link to Class USS_APPLICATION

Application Information

Application Type. . . USS

Job Name. APPSRV06

Application Automation Definition

Job Type. NONMVS

Start Timeout 00:08:00

Startup STARTUP

INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/startappsrv vmlinux6 00 >&
/u/redadm/startappsrv.vmlinux6.00.log'

Startup POSTSTART

INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/checkappsrv vmlinux6 00 >&
/u/redadm/checkappsrv.vmlinux6.00.log'

Shutdown NORM

```

1
INGUSS /bin/kill -2 %PID%

Shutdown FORCE
1
INGUSS /bin/tcsh -c '/u/redadm/stopappsrv vmlinux6 00 >&
/u/redadm/stopappsrv.vmlinux6.00.log'

USS Control
User ID. . . . . REDADM
Command/Path . . . . ./rfcping_vmlinux6_00

```

APPSRV10

This application corresponds to the local application server running on SC42.

The job type of the application is defined as NONMVS because the application server consists of multiple address spaces with identical job names. The starting and stopping of these address spaces is controlled by the application and not by SA OS/390. Therefore, SA OS/390 has to ignore any end-of-job messages; instead, SA OS/390 only monitors the USS process table.

Example 5-28 shows the definition of the application APPSRV10.

Example 5-28 Definition of the application APPSRV10

```

Entry Name: APPSRV10
Link to Class USS_APPLICATION

Application Information
  Application Type. . . USS
  Job Name. . . . . APPSRV10

Application Automation Definition
  Job Type. . . . . NONMVS
  Start Timeout . . . . 00:08:00

Relationships
  Relationship Type . . HASPARENT
  Supporting Resource . TCPIP/APL/=

  Relationship Type . . HASPARENT
  Supporting Resource . RED_DB2GRP/APG/=

  Relationship Type . . HASPARENT
  Supporting Resource . RRS/APL/=

Startup STARTUP

```

```
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/startappsrv wtsc42a 10 >&
/u/redadm/startappsrv.wtsc42a.10.log'
```

Startup POSTSTART

```
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/checkappsrv wtsc42a 10 >&
/u/redadm/checkappsrv.wtsc42a.10.log'
```

Shutdown NORM

```
1
INGUSS /bin/tcsh -c '/u/redadm/stopappsrv wtsc42a 10 >&
/u/redadm/stopappsrv.wtsc42a.10.log'
2
INGUSS /bin/kill -9 %PID%
```

USS Control

```
User ID. . . . . REDADM
Command/Path . . . . ./rfcping_wtsc42a_10
```

APPSRV11

This application corresponds to the local application server running on SC04. Example 5-29 shows the definition of the application APPSRV11.

Example 5-29 Definition of the application APPSRV11

Entry Name: APPSRV11

Link to Class USS_APPLICATION

Application Information

```
Application Type. . . USS
Job Name. . . . . APPSRV11
```

Application Automation Definition

```
Job Type. . . . . NONMVS
Start Timeout . . . . 00:08:00
```

Relationships

```
Relationship Type . . HASPARENT
Supporting Resource . TCPIP/APL/=
```

```
Relationship Type . . HASPARENT
Supporting Resource . RED_DB2GRP/APG/=
```

```
Relationship Type . . HASPARENT
Supporting Resource . RRS/APL/=
```

Startup STARTUP

```
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/startappsrv wtsc04a 11 >&
/u/redadm/startappsrv.wtsc04a.11.log'
```

```

Startup POSTSTART
    INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/checkappsrv wtsc04a 11 >&
    /u/redadm/checkappsrv.wtsc04a.11.log'

Shutdown NORM
    1
    INGUSS /bin/tcsh -c '/u/redadm/stopappsrv wtsc04a 11 >&
    /u/redadm/stopappsrv.wtsc04a.11.log'
    2
    INGUSS /bin/kill -9 %PID%

USS Control
    User ID. . . . . REDADM
    Command/Path . . . . /rfcping_wtsc04a_11

```

Application groups

Having defined the applications, we create an application group to combine the remote application servers (although we have only one remote application server in our configuration). Then we create two superior groups at the sysplex level: one for the remote application servers, and the other for the local application servers.

RED_RASGRP

This application group is created to combine the remote application servers, although we have only one remote application server. Example 5-30 shows the definition of the application group RED_RASGRP.

Example 5-30 Definition of the application group RED_RASGRP

```

Entry Type: ApplicationGroup
Entry Name: RED_RASGRP
Application Group Type . SYSTEM
Nature . . . . . BASIC

Select applications:
    APPSRV06

Relationships
    Relationship Type. . HASPARENT
    Supporting Resource. RED_DB2GRP/APG/=

```

RED_RASPLEX

This application group is a SYSPLEX/MOVE group defined for the remote application servers. These application servers are running on remote systems like UNIX or Windows. They are monitored by SA OS/390 on one only LPAR, as

shown in Figure 5-7 (active applications are represented as shaded boxes). If the LPAR has to be stopped, only the monitoring of the servers is moved via the MOVE group. The application servers themselves will not be stopped.

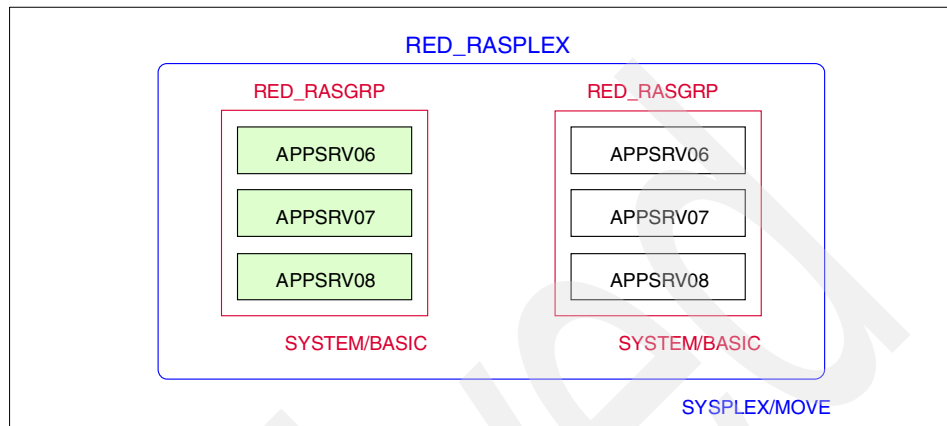


Figure 5-7 RED_RASPLEX application group

Example 5-31 shows the definition of the application group RED_RASPLEX.

Example 5-31 Definition of the application group RED_RASPLEX

```

Entry Type: ApplicationGroup
Entry Name: RED_RASPLEX
Application Group Type . SYSplex
Nature . . . . . MOVE

```

```

Select applications:
  RED_RASGRP/APG/SC04
  RED_RASGRP/APG/SC42

```

RED_LASPLEX

This application group is a SYSplex group defined for the local application servers. One application server is running on each system, as shown on Figure 5-8 on page 113 (active applications are represented as shaded boxes).

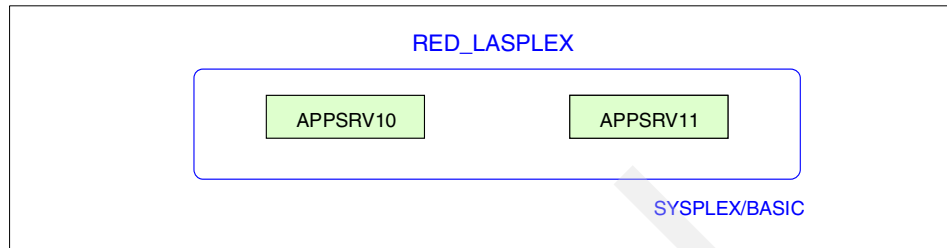


Figure 5-8 RED_LASPLEX application group

Example 5-32 shows the definition of the application group RED_LASPLEX.

Example 5-32 Definition of the application group RED_LASPLEX

```

Entry Type: ApplicationGroup
Entry Name: RED_LASPLEX
Application Group Type . SYSPLEX
Nature . . . . . BASIC

Select applications:
  APPSRV10/APL/SC42
  APPSRV11/APL/SC04
  
```

5.2.7 SAP RED local applications

In this section, we provide the definition of the local applications related to SAP RED: ICLI servers and rfcoscol. These applications are started on every LPAR on which RED SAP is running.

Applications

We define one application for each ICLI server (we defined four ICLI servers, and therefore four applications, but we only document the definition of REDICLI6), and one application for rfcoscol.

REDICLI6

This application corresponds to the ICLI server used by APPSRV06 to connect to the database server.

Example 5-33 on page 114 shows the definition of the application REDICLI6. As we have chosen to start the ICLI servers via a start procedure, this application is defined as a STANDARD application.

Example 5-33 Definition of the application REDICLI6

Entry Name:REDICLI6

Application Information

Application Type. . . STANDARD
Job Name. REDICLI6

Relationships

Relationship Type . . HASPPARENT
Supporting Resource . RED_DB2GRP/APG/=

Relationship Type . . HASPPARENT
Supporting Resource . RRS/APL/=

Shutdown NORM

1
MVS F &SUBSJOB,APPL=STOP TIMEOUT(60)
2
MVS P &SUBSJOB
3
MVS C &SUBSJOB

RED_RFC

This application corresponds to the rfcscol. Example 5-34 shows the definition of the application RED_RFC.

Example 5-34 Definition of the application RED_RFC

Entry Name: RED_RFC

Link to Class USS_APPLICATION

Application Information

Application Type. . . USS
Job Name. REDADMRI

Relationships

Relationship Type . . HASPPARENT
Supporting Resource . RED_DB2GRP/APG/=

Relationship Type . . HASPPARENT
Supporting Resource . RRS/APL/=

Relationship Type . . HASPPARENT
Supporting Resource . SAP_OSCOL/APL/=

Startup

INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/u/redadm/start_rfcscol >&
/u/redadm/rfcscol.&SYSNAME..log'

```

Shutdown NORM
1
INGUSS /bin/kill -9 %PID%

USS Control
User ID. . . . . REDADM
Command/Path . . . . /usr/sap/RED/SYS/exe/run/rfcoscol

```

Application group

Having defined the applications, we create an application group to combine the local application related to SAP RED.

RED_LOCAL

This SYSTEM group combines the SAP RED local applications running on a single system, as shown on Figure 5-9 (active applications are represented as shaded boxes).

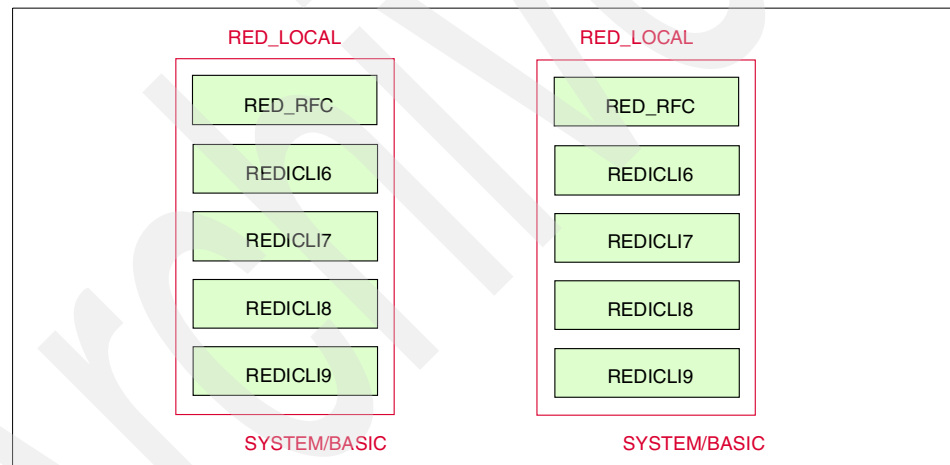


Figure 5-9 RED_LOCAL application group

Example 5-35 shows the definition of the application group RED_LOCAL.

Example 5-35 Definition of the application group RED_LOCAL

```

Entry Type: ApplicationGroup
Entry Name: RED_LOCAL
Application Group Type . SYSTEM
Nature . . . . . BASIC

```

```

Select applications:
  RED_RFC

```

REDICLI6
REDICLI7
REDICLI8
REDICLI9

Relationships

Relationship Type. . HASPARENT
Supporting Resource. TCPIP/APL/=

5.2.8 NFS server

In this section, we provide the definition of the NFS server.

Application

We define one application for the NFS server.

MVSNFSSA

This application corresponds to the NFS server. Example 5-36 shows the definition of the application MVSNFSSA.

Example 5-36 Definition of the application MVSNFSSA

Entry Name: MVSNFSSA

Application Information

Application Type. . . STANDARD
Job Name. MVSNFSSA
JCL Procedure Name. . MVSNFSSA

Relationships

Relationship Type . . MAKEAVAILABLE
Supporting Resource . NFS_HAPLEX/APG
Automation. PASSIVE
Chaining. WEAK
Condition WhenObservedDown

Relationship Type . . MAKEAVAILABLE
Supporting Resource . TCPIP/APL/=

Automation. ACTIVE
Chaining. WEAK
Condition WhenAvailable

Startup POSTSTART

MVS SETOMVS FILESYS,FILESYSTEM='SAPRED.SHFS.SAPMNT',SYSNAME=&SYSNAME.
MVS SETOMVS FILESYS,FILESYSTEM='SAPRED.SHFS.TRANS',SYSNAME=&SYSNAME.

Shutdown NORM

```
1
MVS P &SUBSJOB
4
MVS C &SUBSJOB
```

Application group

We create one application group at the sysplex level.

NFS_HAPLEX

The NFS server should run on one of the two systems at a time. Therefore, we define a SYSPLEX/MOVE group with the NFS server as the only member, as shown in Figure 5-10 (active applications are represented as shaded boxes).

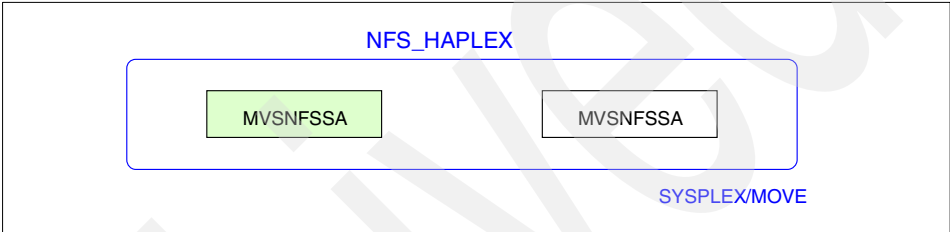


Figure 5-10 NFS_HAPLEX application group

Example 5-37 shows the definition of the application group NFS_HAPLEX.

Example 5-37 Definition of the application group NFS_HAPLEX

```
Entry Type: ApplicationGroup
Entry Name: NFS_HAPLEX
Application Group Type . SYSPLEX
Nature . . . . . MOVE

Select applications:
  MVSNFSSA

Relationships
  Relationship Type. . HASPARENT
  Supporting Resource. TCPIP/APL/=
```

5.2.9 saprouter

In this section, we describe the definition of the saprouter.

Applications

We define two applications: one for the VIPA associated with the saprouter, and the other one for the saprouter itself.

SAP_RTVIPA

This application corresponds to the VIPA associated with the saprouter.

Example 5-38 shows the definition of the application SAP_RTVIPA.

Example 5-38 Definition of the application SAP_RTVIPA

Entry Name: SAP_RTVIPA

Application Information

Application Type. . . STANDARD
Job Name. TCPVIPAR
JCL Procedure Name. . TCPVIPA

Application Automation Definition

Job Type. TRANSIENT
Transient Rerun . . . YES

Startup

Parameters. ,VIPA='172.20.10.3'

SAP_ROUTER

This application corresponds to the saprouter. Example 5-39 shows the definition of the application SAP_ROUTER.

Example 5-39 Definition of the application SAP_ROUTER

Entry Name: SAP_ROUTER

Link to Class USS_APPLICATION

Application Information

Application Type. . . USS
Job Name. SAPROUTE

Relationship Type. . HASPARENT

Supporting Resource. SAP_RTVIPA/APL/=

Startup

INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/usr/sap/RED/SYS/exe/run/saprouter -r
>& /u/redadm/start_subrouter.&SYSNAME..log'

Shutdown NORM

1
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/usr/sap/RED/SYS/exe/run/saprouter -s
>& /u/redadm/stop_subrouter.&SYSNAME..log'

```
USS Control
  User ID. . . . . REDADM
  Command/Path . . . . /usr/sap/RED/SYS/exe/run/saprouter
```

Application group

Having defined the applications, we create an application group to combine them at the system level. Then, we create a superior group at the sysplex level.

SAP_RTGRP

The SAP router and its associated VIPA must run together on the same LPAR. Therefore, we group them together in a SYSTEM group.

Example 5-40 shows the definition of the application group SAP_RTGRP.

Example 5-40 Definition of the application group SAP_RTGRP

```
Entry Type: ApplicationGroup
Entry Name: SAP_RTGRP
Application Group Type . SYSTEM
Nature . . . . . BASIC

Select applications:
  SAP_ROUTER
  SAP_RTVIPA

Relationships
  Relationship Type. . HASPARENT
  Supporting Resource. TCPIP/APL/=
```

SAP RTPLEX

The saprouter (and its associated VIPA) should run on one of the two systems at a time. Therefore, we define a SYSPLEX/MOVE application group, as shown on Figure 5-11 on page 120 (active applications are represented as shaded boxes).

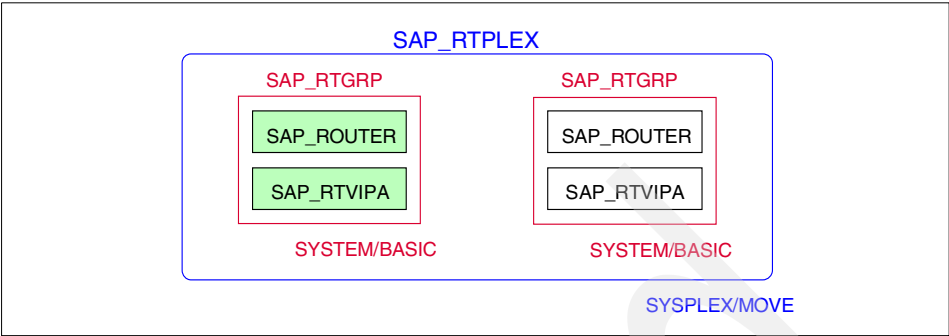


Figure 5-11 SAP RTPLEX application group

Example 5-41 shows the definition of the application group SAP RTPLEX.

Example 5-41 Definition of the application group SAP RTPLEX

Entry Type: ApplicationGroup
Entry Name: SAP RTPLEX
Application Group Type . SYSPLEX
Nature MOVE

Select resources:
SAP_RTGRP/APG/SC04
SAP_RTGRP/APG/SC42

Relationships
Relationship Type. . HASPARENT
Supporting Resource. TCPIP/APL/=

5.2.10 SAP local application

In this section, we describe the definition of the SAP local application saposcol. This application is started once on every system on which an SAP is running.

Application

We define one application for saposcol.

SAP_OSCOL

This application corresponds to saposcol. Example 5-42 on page 121 shows the definition of the application SAP_OSCOL.

Example 5-42 Definition of the application SAP_OSCOL

Entry Name: SAP_OSCOL
Link to Class USS_APPLICATION

Application Information
Application Type. . . USS
Job Name. REDADMOS

Startup
INGUSS JOBNAME=&SUBSJOB,/bin/tcsh -c '/usr/sap/RED/SYS/exe/run/saposcol -l
>& /u/redadm/saposcol.&SYSNAME..log'

Shutdown NORM
1
INGUSS /bin/kill -2 %PID%
4
INGUSS /bin/kill -9 %PID%

USS Control
User ID. REDADM
Command/Path /usr/sap/RED/SYS/exe/run/saposcol

Application group

We create one application group to combine the SAP local application (although, in our case, we have only one SAP local application saposcol).

SAP_LOCAL

This group, as shown in Figure 5-12, combines applications running on each LPAR. In fact, in our environment, this is just the application SAP_OSCOL (active applications are represented as shaded boxes).

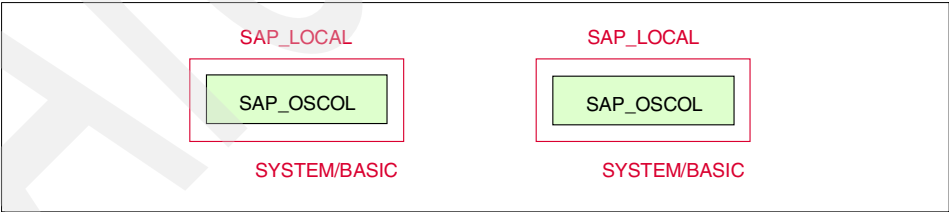


Figure 5-12 SAP_LOCAL application group

Example 5-43 on page 122 shows the definition of the application group SAP_LOCAL.

Example 5-43 Definition of the application group SAP_LOCAL

Entry Type: ApplicationGroup
Entry Name: SAP_LOCAL
Application Group Type . SYSTEM
Nature BASIC

Select applications:
SAP_OSCOL

5.2.11 Defining superior groups

We define two superior SYSPLEX application groups to combine the SAP-related resources together. These groups will serve as the entry point for monitoring and operations.

RED_SAPPLEX

This SYSPLEX application group combines all resources that belong to the SAP system RED, as shown on Figure 5-13.

Tip: If you configure more than one SAP system, you should define such a superior group for each one of them.

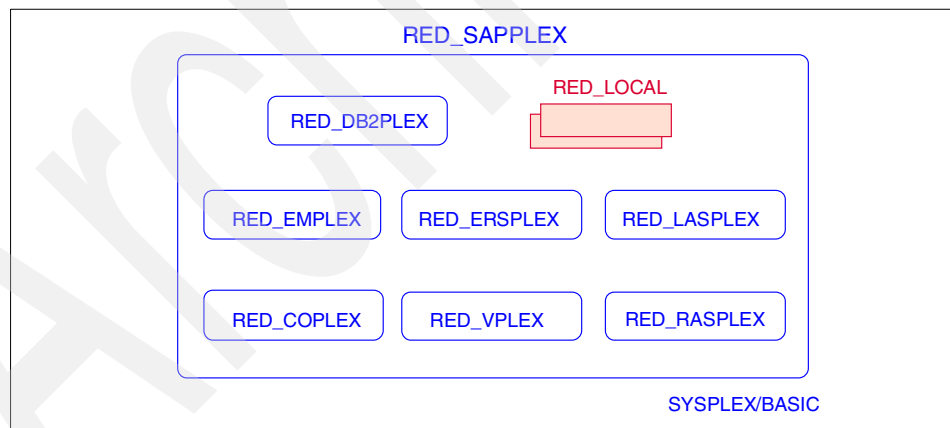


Figure 5-13 RED_SAPPLEX application group

Example 5-44 on page 123 shows the definition of the application group RED_SAPPLEX.

Example 5-44 Definition of the application group RED_SAPPLEX

```
Entry Type: ApplicationGroup
Entry Name: RED_SAPPLEX
Application Group Type . SYSPLEX
Nature . . . . . BASIC
```

```
Select resources:
  RED_COPLEX/APG
  RED_DB2PLEX/APG
  RED_EMPLX/APG
  RED_ERSPLEX/APG
  RED_LASPLEX/APG
  RED_LOCAL/APG/SC04
  RED_LOCAL/APG/SC42
  RED_RASPLEX/APG
  RED_VPLEX/APG
```

SAP

This SYSPLEX application group is the top level group of all SAP-related resources, as shown on Figure 5-14.

Tip: This group is also very useful when using the Status Display Facility (SDF). Define SAP as an active symbol on the SDF screen and it will change color on every status change of any SAP-related resource in the sysplex.

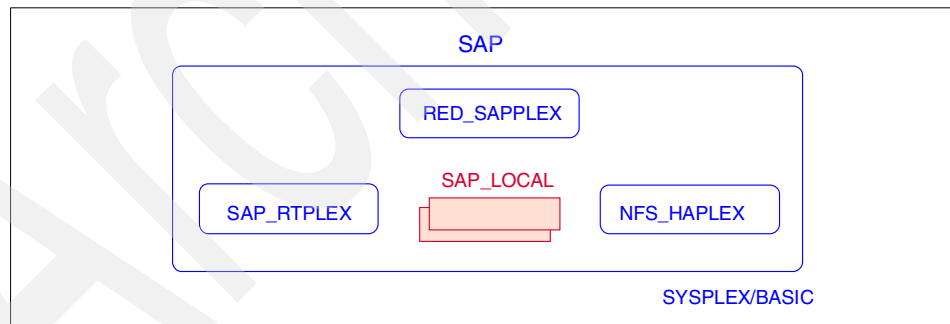


Figure 5-14 SAP application group

Example 5-45 shows the definition of the application group SAP.

Example 5-45 Definition of the application group SAP

Entry Type: ApplicationGroup
Entry Name: SAP
Application Group Type . SYSPLEX
Nature BASIC

Select resources:
NFS_HAPLEX/APG
RED_SAPPLEX/APG
SAP_LOCAL/APG/SC04
SAP_LOCAL/APG/SC42
SAP_RTPLEX/APG

5.3 Overall picture

Figure 5-15 gives you the overall picture of all of the groups and applications that we defined in our SA OS/390 policy database.

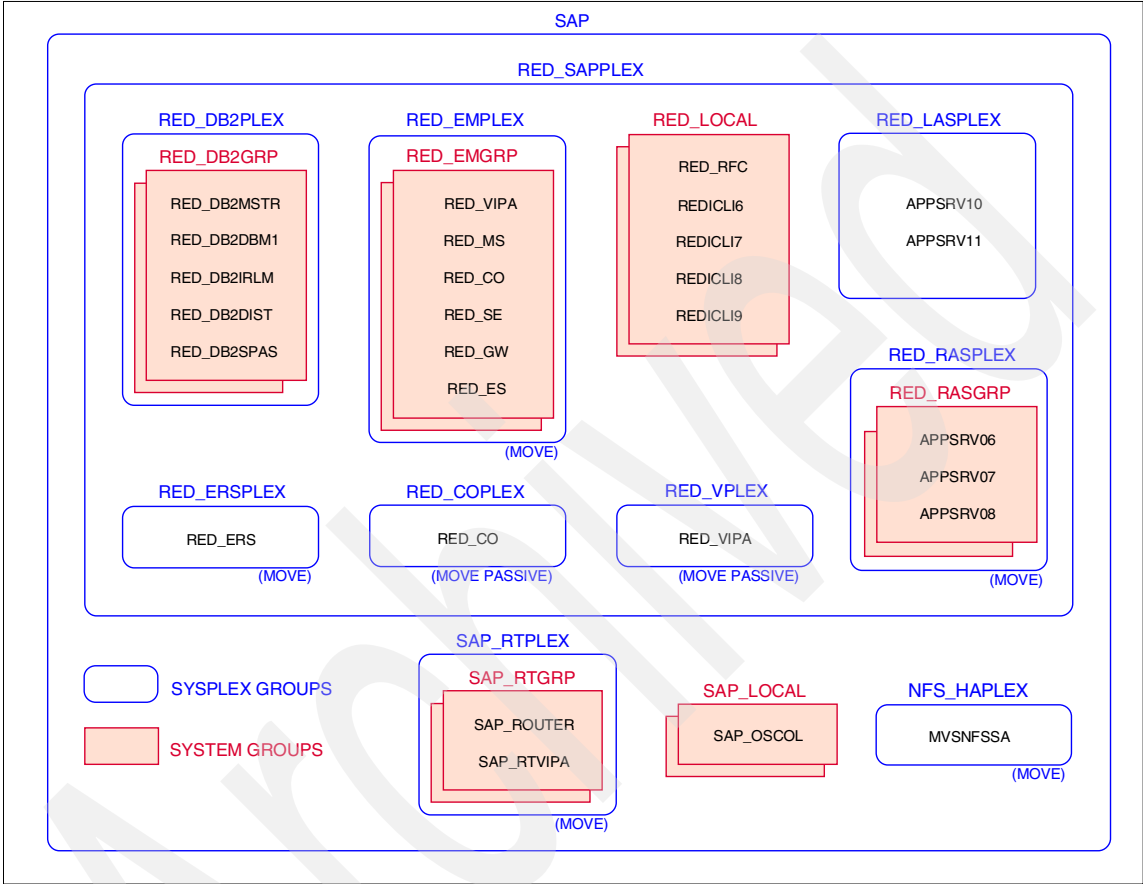


Figure 5-15 Overview of the resources

5.4 Summary tables

The following tables summarize all groups and applications we defined, with a short description, and the page on which you can find the parameters to be entered in the SA OS/390 policy.

5.4.1 Classes

Table 5-2 summarizes all the classes we used.

Table 5-2 Summary of the classes

Name	Description	Page
CLASS_DB2_MASTER	Class for DB2 MSTR address space	91
CLASS_RED_DB2_CHILDS	Class for SAP RED DB2 children	92
USS_APPLICATION	Class for all USS applications	90

5.4.2 Applications

Table 5-3 summarizes all the applications we defined.

Table 5-3 Summary of the applications

Name	Description	Page
APPSRV06	RED SAP application server on VMLINUX6 (remote)	108
APPSRV10	RED SAP application server on SC42 (local)	109
APPSRV11	RED SAP application server on SC04 (local)	110
MVSNFSSA	Network File System server for TCPIPA	116
RED_DB2DBM1	RED DB2 DBM1 address space	94
RED_DB2DIST	RED DB2 DIST address space	94
RED_DB2IRLM	RED DB2 IRLM address space	94
RED_DB2MSTR	RED DB2 MSTR address space	93
RED_DB2SPAS	RED DB2 SPAS address space	95
RED_CO	RED SAP syslog collector	99
RED_ERS	RED SAP enqueue replication server	102
RED_ES	RED SAP enqueue server	97
RED_GW	RED SAP gateway	101
RED_MS	RED SAP message server	99
RED_RFC	RED SAP rfcoscol	114
RED_SE	RED SAP syslog sender	100

Name	Description	Page
RED_VIPA	VIPA related to RED SAP enqueue instance	101
REDICLI6	ICLI server for APPSRV6	113
SAP_OSCOL	saposcol, runs once for all SAPs on one LPAR	120
SAP_ROUTER	saprouter	118
SAP_RTVIPA	VIPA related to saprouter	118

5.4.3 Application groups

Table 5-4 summarizes all application groups we defined.

Table 5-4 Summary of the application groups

Name	Type	Description	Page
NFS_HAPLEX	SYSPLEX MOVE	All MVSNFSSA applications	117
RED_COPLEX	SYSPLEX MOVE PASSIVE	All RED_CO applications	107
RED_DB2GRP	SYSTEM BASIC	All RED_DB2* applications	95
RED_DB2PLEX	SYSPLEX BASIC	All RED_DB2GRP application groups	96
RED_EMGRP	SYSTEM BASIC	All enqueue instance components	104
RED_EMPLX	SYSPLEX MOVE	All RED_EMGRP application groups	104
RED_ERSPLEX	SYSPLEX MOVE	All RED_ERS applications	104
RED_LASPLEX	SYSPLEX BASIC	All local application servers	112
RED_LOCAL	SYSTEM BASIC	All REDICLI* + RED_RFC applications	115
RED_RASGRP	SYSTEM BASIC	All remote application servers	111
RED_RASPLEX	SYSPLEX MOVE	All RED_RASGRP application groups	111
RED_SAPPLEX	SYSPLEX BASIC	All resources belonging to SAP RED	122
RED_VPLEX	SYSPLEX MOVE PASSIVE	All RED_VIPA applications	106
SAP	SYSPLEX BASIC	All bits and pieces of SAP	123
SAP_LOCAL	SYSTEM BASIC	All SAP_OSCOL applications	121
SAP_RTGRP	SYSTEM BASIC	All SAP_ROUTER + SAP_RTVIPA applications	119
SAP_RTPLEX	SYSPLEX MOVE	All SAP_RTGRP application groups	119

5.5 Extension for DFS/SMB

This is an extension to 5.2, “Defining the SAP-related resources” on page 89. We describe here how to add the definitions for DFS/SMB to the SA OS/390 policy, to SDF, and to the Automation Table.

5.5.1 Additions to the SA OS/390 policy

In this section, we provide the additions to the SA OS/390 policy.

Application

We define one application for DFS/SMB.

DFS_SMB

This application corresponds to DFS_SMB. Example 5-46 shows the definition of the application DFS_SMB.

Example 5-46 Definition of the application DFS_SMB

Entry Name: DFS_SMB

Application Information

Application Type. . . STANDARD
Job Name. DFS_SMB
JCL Procedure Name. . DFS

Relationships

Relationship Type . . MAKEAVAILABLE
Supporting Resource . SMB_PLEX/APG
Automation. PASSIVE
Chaining. WEAK
Condition WhenObservedDown

Relationship Type . . MAKEAVAILABLE
Supporting Resource . TCPIP/APL/=
Automation. ACTIVE
Chaining. WEAK
Condition WhenAvailable

PRESTART

MVS SETOMVS FILESYS,FILESYSTEM='SAPRED.SHFS.SAPMNT',SYSNAME=&SYSNAME.
MVS SETOMVS FILESYS,FILESYSTEM='SAPRED.SHFS.TRANS',SYSNAME=&SYSNAME.

Shutdown NORM

1
MVS P &SUBSJOB
4
MVS C &SUBSJOB

Application group

We define one application group for DFS/SMB.

SMB_PLEX

DFS/SMB should run on one of the two systems at a time. Therefore, we define a SYSPLEX/MOVE group with DFS/SMB, as shown in Figure 5-16 (active applications are represented as shaded boxes).

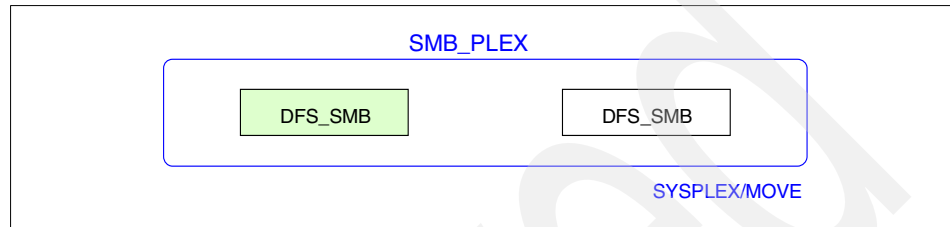


Figure 5-16 SMB_PLEX application group

Example 5-47 shows the definition of the application group SMB_PLEX.

Example 5-47 Definition of the application group SMB_PLEX

```
Entry Type: ApplicationGroup
Entry Name: SMB_PLEX
Application Group Type . SYSPLEX
Nature . . . . . MOVE
```

```
Select applications:
  DFS_SMB
```

We want to have both subsystems MVSNFSSA and DFS_SMB always running on the same LPAR, and want to always move them together; this is why we insert the following STARTUP POSTSTART commands:

For MVSNFSSA:

```
INGGROUP SMB_PLEX/APG,ACTION=RESET
INGGROUP SMB_PLEX/APG,ACTION=ADJUST,MEMBERS=(DFS_SMB/APL/&SYSNAME.), PREF=(999)
```

For DFS_SMB:

```
INGGROUP NFS_HAPLEX/APG,ACTION=RESET
INGGROUP NFS_HAPLEX/APG,ACTION=ADJUST,MEMBERS=(MVSNFSSA/APL/&SYSNAME.),
PREF=(999)
```

If DFS_SMB moves to a different LPAR, the POSTSTART command of DFS_SMB first resets the preference value of the NFS_HAPLEX group to default. Then, it sets the preference value for MVSNFSSA to 999.

This will cause MVSNFSSA to move also to the LPAR on which DFS_SMB is restarted, since the running MVSNFSSA application has a preference value of only 950.

5.5.2 Additions to SDF

We add the following entries for DFS_SMB to our sample SDF panel AOFSAP; see Example 5-48.

Example 5-48 Additions to our SDF panel AOFSAP

```
SF(SC04.DFS_SMB,07,40,52,N, ,)
ST(DFS_SMB      )
SF(SC42.DFS_SMB,07,54,66,N, ,)
ST(DFS_SMB      )
```

An entry DFS_SMB is also added to the SDF tree. One extra line is inserted in the members AOFTSC04 and AOFTSC42 as shown in Example 5-49.

Example 5-49 Additions to the SDF tree

```
...
010700  2  SAP
010800  3  MVSNFSSA
010810  3  DFS_SMB
010900  3  SAP_ROUTER
...
```

5.5.3 Additions to the Automation Table

We define IOEPO1103I as the UP message and IOEPO1100I as the DOWN message for the DFS subsystem, as shown in Example 5-50 on page 131.

Example 5-50 Additions to the automation table

```
*****
*
* DFS
*
* -----*
*
* IOEP01103I DFS KERNEL INITIALIZATION COMPLETE. ==> UP MESSAGE
*
* IOEP01100I DFS DAEMON DFSKERN HAS STOPPED. ==> FINAL END MESSAGE
*
*****
*
IF MSGID = 'IOEP' . & DOMAINID = '&DOMAIN.' THEN BEGIN;
*
    IF MSGID = 'IOEP01103I' .
        THEN EXEC(CMD('ACTIVMSG UP=YES'))
            ROUTE(ONE %AOFOPGSSOPER%);
*
    IF MSGID = 'IOEP01100I' .
        THEN EXEC(CMD('TERMMSG FINAL=YES'))
            ROUTE(ONE %AOFOPGSSOPER%);
*
    ALWAYS;
```



Verification procedures and failover scenarios

This chapter describes the test scenarios we designed and ran to test the SA OS/390 policy.

6.1 Overview of the test scenarios

Before defining and running test scenarios to verify the SA OS/390 policy, we made the following assumptions:

- ▶ The z/OS and network configuration was done.
- ▶ The SAP HA installation was done.
- ▶ The SA OS/390 and NetView configuration was done.
- ▶ The complete environment is available.

6.1.1 Classification of the test scenarios

The scenarios must cover both *planned outages* (or planned activities) and *unplanned outages* (or failures). And for each category, tests must be run at the *component* level (the component can be related to SAP, z/OS, or the network) and at the *LPAR* level.

Table 6-1 depicts, in the form of a matrix, some examples of test scenarios.

Table 6-1 Examples of test scenarios

	Planned outages	Unplanned outages
Component	<ul style="list-style-type: none">▶ Shutdown of a DB2 subsystem for maintenance▶ Stop of an SAP application server for kernel upgrade	<ul style="list-style-type: none">▶ Failure of a TCP/IP stack▶ Failure of the enqueue server
LPAR	<ul style="list-style-type: none">▶ Shutdown of an LPAR for hardware upgrade▶ Shutdown of an LPAR for re-IPLing	<ul style="list-style-type: none">▶ Power outage▶ Unrecoverable operating system failure

6.1.2 Test scenarios to verify the SA OS/390 policy

We built a list of test scenarios, including planned and unplanned outages, to verify the SA OS/390 policy.

Planned outage scenarios

- ▶ Controlled operator intervention against SAP-related components:
 - Start and stop of all the SAP-related components
 - Start and stop of the entire SAP RED system
 - Start and stop of the enqueue instance

- Move of the enqueue instance from one LPAR to the other
- Start and stop of the enqueue replication server
- Move of the enqueue replication server from one LPAR to another (if more than two LPARs)
- Start and stop of the enqueue server
- Start and stop of the message server
- Start and stop of the NFS server
- Move of the NFS server from one LPAR to the other
- Start and stop of all DB2 subsystems belonging to the SAP system
- Start and stop of a single DB2 subsystem
- Start and stop of an application server on z/OS
- Start and stop of an application server on Linux for zSeries
- ▶ Startup of the entire sysplex:
 - Startup of all LPARs one after the other
- ▶ Planned shutdown and restart of an LPAR containing SAP critical components:
 - Shutdown and restart of the LPAR where the enqueue server and the NFS server are running
 - Shutdown and restart of the LPAR where the enqueue replication server is running

Unplanned outage scenarios

- ▶ Failure of an SAP component:
 - The enqueue server
 - The enqueue replication server
 - The message server
 - An ICL1 server
 - An application server on z/OS
 - An application server on Linux for zSeries
 - A DB2 subsystem
 - The NFS server
 - The syslog collector
 - A syslog sender

- The SAP gateway
- Saprouter
- Saposcol
- Rfcoscol
- ▶ Failure of a network component:
 - A TCP/IP stack on z/OS
 - OSPF (OMPROUTE)
 - A network adapter on zSeries
 - A network switch
- ▶ Failure of an LPAR:
 - The LPAR where the enqueue replication server is running
 - The LPAR where the enqueue server and the NFS server are running

6.1.3 Scenarios that we tested

During this residency, we tested the following scenarios:

Planned outage scenarios

- ▶ Controlled operator intervention against SAP-related components:
 - Start and stop of the entire SAP RED system
- ▶ Startup of the entire sysplex:
 - Startup of all LPARs, one after the other
- ▶ Planned shutdown and restart of an LPAR containing critical SAP components:
 - Shutdown and restart of the LPAR where the enqueue server and the NFS server are running

Unplanned outage scenarios

- ▶ Failure of a critical SAP component:
 - The enqueue server
 - The message server
 - An ICL server

- ▶ Failure of a critical network resource:
 - The NFS server
 - A TCP/IP stack
- ▶ Failure of an LPAR containing critical SAP components:
 - The LPAR where the enqueue server and NFS server are running

6.2 Test methodology

Although each scenario is different, many of the steps that need to be executed before, during, and after the test are similar. We describe these steps in the following section in the form of a methodology that we followed all through our tests, and which you can apply for any scenario you may want to test in your own environment.

6.2.1 Purpose of the test

We characterize the purpose of the test with two points:

- ▶ The *scope* of the test: Is the test run against a single component (for example, the enqueue server), a group of resources (for example, the whole SAP system), or an entire LPAR?
- ▶ The *action* to be tested: Do we want to test a normal startup or shutdown, a controlled movement, or do we want to simulate a failure?

6.2.2 Expected behavior

We describe the expected behavior of every component impacted during the test: Should it stop, restart in the same LPAR, move to the other LPAR, what should happen to the application servers, what about transparency for the running workload?

6.2.3 Setup of the test environment

We prepare the test environment knowing which resources must be stopped, which must be up, and in which LPAR each component must be running.

6.2.4 Verification of resource status

Before each test, we used the following checklist to review the status of all the SAP-related resources defined in SA OS/390:

1. Do all the resources monitored by SA OS/390 have a compound status SATISFACTORY?

Tip: The NetView command INGLIST SAP/APG displays the status of the application group SAP. If the compound status is SATISFACTORY, then we know that all resources belonging to that group have a compound state SATISFACTORY. Otherwise, we can drill down the tree of resources using option G (Members).

Figure 6-1 is a sample output of the NetView command INGLIST SAP/APG, showing the application group SAP with a compound status of SATISFACTORY.

INGKYSTO		SA OS/390 - Command Dialogs		Line 1 of 1	
Domain ID = SC04A		----- INGLIST -----		Date = 06/03/02	
Operator ID = NETOP1		Sysplex = WTSCPLX1		Time = 16:04:34	
CMD: A Update		B Start	C Stop	D INGRELS	E INGVOTE
G Members		H DISPTRG	I INGSCHED	J INGGROUP	F INGINFO
CMD Name		Type System	Compound	Desired	Observed Nature
-----		-----			
SAP		APG	SATISFACTORY	AVAILABLE	AVAILABLE BASIC

Figure 6-1 Sample output of the NetView command INGLIST

2. Are there any outstanding votes in SA OS/390?

Tip: The NetView command INGVOTE displays the list of all the votes in the system. The list should be empty.

Figure 6-2 is a sample output of the NetView command INGVOTE, showing that there are no outstanding votes.

INGKYRQ2		SA OS/390 - Command Dialogs		Line 1 of 5	
Domain ID = SC04A		----- INGVOTE -----		Date = 06/03/02	
Operator ID = NETOP1		Sysplex = WTSCPLX1		Time = 16:24:31	
Cmd: C Cancel request		K Kill request	S Show details	V Show votes	
Cmd Name		Type System	Request Data		

Figure 6-2 Sample output of the NetView command INGVOTE

3. Are there any outstanding excludes in SA OS/390?

Note: There is no command to display all the excludes in SA OS/390 at once. Individual INGINFO commands must be issued against every application group defined as SYSPLEX/MOVE groups.

In our configuration, we used the following commands:

```
INGINFO RED_EMPLX
INGINFO RED_ERSPLEX
INGINFO NFS_HAPLEX
INGINFO RED_RASPLEX
INGINFO SAP_RTPLEX
INGINFO RED_COPLX
INGINFO RED_VPLX
```

Figure 6-3 shows a sample output of the NetView command INGINFO. We look more specifically at the section Group Details (on the third screen of the display). It shows that SC42 is in the exclude list of the application group RED_EMPLX.

```
INGKYINO          SA OS/390 - Command Dialogs          Line 43 of 189
Domain ID   = SC42A          ----- INGINFO -----      Date = 06/06/02
Operator ID = NETOP2          Sysplex = WTSCPLX1          Time = 11:03:14

Resource ==> RED_EMPLX/APG          format: name/type/system
System   ==>          System name, domain ID or sysplex name

Group Details...
Nature    : MOVE
Members   :
  RED_EMGRP/APG/SC04          Enqueue Group
    PREF = 700
    PREFADJ = 0
    SYSTEMS = SC04
  RED_EMGRP/APG/SC42          Enqueue Group
    PREF = 700
    PREFADJ = 0
    SYSTEMS = SC42

Policy    :
  PASSIVE = NO
  EXCLUDE = SC42
```

Figure 6-3 Sample output of the NetView command INGINFO

We usually do not want any excludes before the test. Therefore, this exclude should be removed by issuing the NetView command INGGROUP, as shown:

```
INGGROUP RED_EMPLX/APG ACTION=INCLUDE SYSTEMS=SC42
```

Tip: Instead of seven INGINFO commands, we used a “home-developed” REXX procedure called SANCHK to display and remove all the outstanding excludes in SA OS/390. The code source for this procedure can be found in D.2.1, “SANCHK” on page 266.

Figure 6-4 shows the output of the REXX procedure SANCHK. It shows that we have two outstanding excludes: SC42 is in the exclude list of the application groups RED_EMPLX and NFS_HAPLEX.

```
* SC04A  SANCHK
| SC04A  Gathering data step 1 ...
| SC04A  Gathering data step 2 ...
| SC04A  Nothing to display ...
* SC04A  SANCHK
| SC04A  Gathering data step 1 ...
| SC04A  Gathering data step 2 ...
| SC04A

-----
Group      = NFS_HAPLEX/APG
Excluded   = SC42
Avoided    =

-----
Group      = RED_EMPLX/APG
Excluded   = SC42
Avoided    =

-----
End of Sanity Check
```

Figure 6-4 Displaying all the outstanding excludes using SANCHK

We can also use the REXX procedure SANCHK with the option CLEAR to remove all the excludes; see Figure 6-5.

```
* SC04A  SANCHK CLEAR
| SC04A  Gathering data step 1 ...
| SC04A  Gathering data step 2 ...
| SC04A  Processing CLEAR ...
| SC04A  Processing CLEAR for NFS_HAPLEX/APG
U SC04A  AOF099I FUNCTION SUCCESSFULLY COMPLETED
| SC04A  Processing CLEAR for RED_EMPLX/APG
U SC04A  AOF099I FUNCTION SUCCESSFULLY COMPLETED
| SC04A  Finished CLEAR processing
```

Figure 6-5 Removing all the outstanding excludes using SANCHK

4. Where are the enqueue server, message server, enqueue replication server and NFS server running before the test?

Tip: We customized an SDF panel to monitor all the SAP-related resources and to see on which system they are running (for more information, refer to 5.1.5, “Sending UNIX messages to the syslog” on page 89).

Figure 6-6 is a sample screen showing, on the left-hand side, the SAP-related components that are associated with each system. On the right-hand side, it shows the SAP-related components that can be moved from one system to the other.

In this example, the enqueue server (RED_ES), the NFS server (MVSNFSSA), and saprouter (SAP_ROUTER) are running on SC04, and the enqueue replication server (RED_ERS) is running on SC42.

S A P High Availability					
Local Applications		Moving Applications			
SC04	SC42	SC04	SC42		
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA		
RED_DB2DBM1	RED_DB2DBM1				
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA		
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER		
RED_DB2SPAS	RED_DB2SPAS				
		RED_VIPA	RED_VIPA		
RED_RFC	RED_RFC	RED_ES	RED_ES		
REDICLI6	REDICLI6	RED_MS	RED_MS		
REDICLI7	REDICLI7	RED_GW	RED_GW		
REDICLI8	REDICLI8	RED_CO	RED_CO		
REDICLI9	REDICLI9	RED_SE	RED_SE		
		RED_ERS	RED_ERS		
APPSRV11	APPSRV10	APPSRV06	APPSRV06		
SAP_OSCOL	SAP_OSCOL	APPSRV07	APPSRV07		
		APPSRV08	APPSRV08		
06/06/02 13:40					

Figure 6-6 Sample SDF customized panel

5. Are the NFS file systems mounted on the remote application server accessible?

Tip: We either logon to the remote application server and display the available file systems (using the UNIX command `df`), or we use the SAP transaction AL11 to check that we can access the files in the SAP directories.

6.2.5 Preparation for the test (unplanned outage only)

During the unplanned outage scenarios, we want to verify the impact of the failure for end users and for any workload that would be running on the system. Therefore, before each test, we execute the following preparation steps:

1. Log on to all the application servers.
2. Create an SAP workload.

Tip: To generate a workload you may use, for example, home-written batch jobs, or start a client copy.

We used a workload generated by a solution test tool called ZAP1. The program goes through an insert/update/delete cycle several times. We set a sleep time between every step. During sleep time, the current work process is released (to be available for other tasks). After sleep time, the program gets a work process again and continues with the next step. Our workload consisted of five of these programs running in parallel.

3. Generate entries in the enqueue table.

Tip: We use transaction SM12 to generate entries in the enqueue table.

From the primary panel of transaction SM12, Select Lock Entries, enter test in the transaction field, as shown on Figure 6-7.

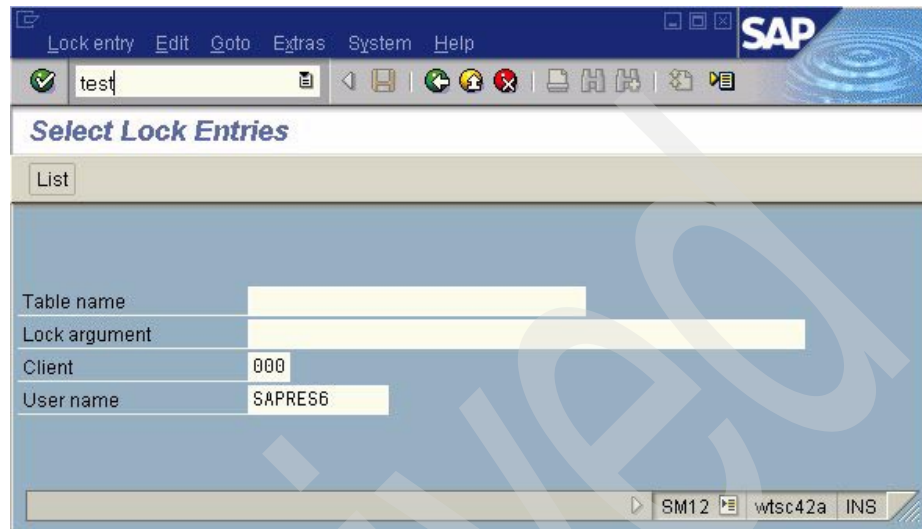


Figure 6-7 SM12 primary panel

A new selection appears in the menu bar: "Error handling".

Click **Error handling** → **Test tools** → **Mass calls** (see Figure 6-8).

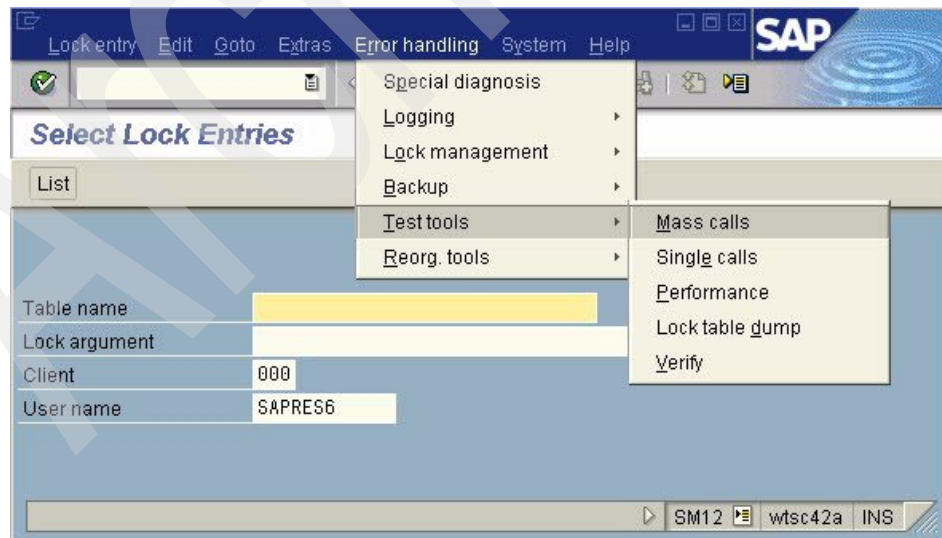


Figure 6-8 Error handling menu

Choose the number of lock entries you want to create (for our test purposes, we always used the default of 10 lock entries), then click **Execute**; see Figure 6-9.

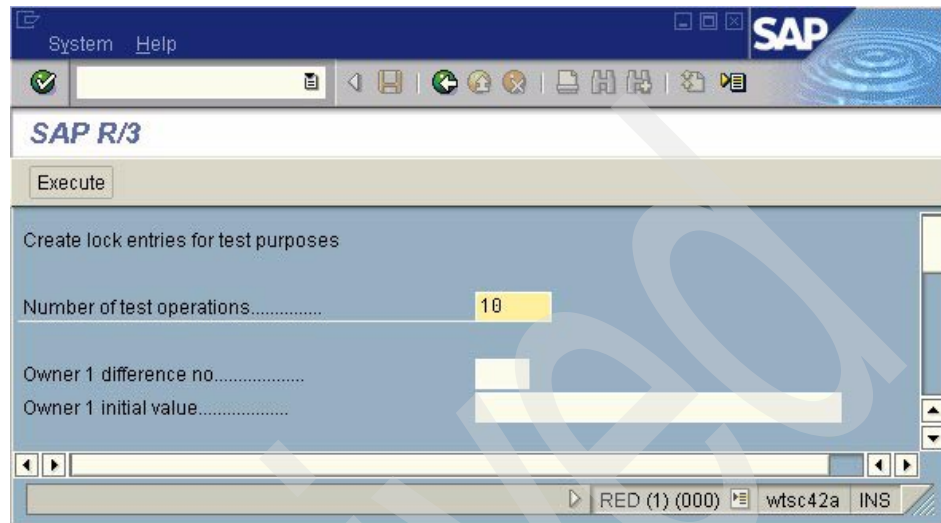


Figure 6-9 Enqueue test: start mass enqueue operations

The screen must stay open for the duration of the test. From *another* screen, we use SM12 to list the entries in the enqueue table; see Figure 6-10 on page 145.

C11 User	Time	Shared Table	Lock argument
000 SAPRES6	15:28:01	GRA 0	ARG 0
000 SAPRES6	15:28:01	GRA 1	ARG 1
000 SAPRES6	15:28:01	GRA 2	ARG 2
000 SAPRES6	15:28:01	GRA 3	ARG 3
000 SAPRES6	15:28:01	GRA 4	ARG 4
000 SAPRES6	15:28:01	GRA 5	ARG 5
000 SAPRES6	15:28:01	GRA 6	ARG 6
000 SAPRES6	15:28:01	GRA 7	ARG 7
000 SAPRES6	15:28:01	GRA 8	ARG 8
000 SAPRES6	15:28:01	GRA 9	ARG 9

Selected lock entries: 10

Figure 6-10 List of the entries in the enqueue table

6.2.6 Execution of the test

The initiation of the test depends on the type of scenario.

- ▶ For a planned outage or a controlled move of resources, SA OS/390 must be used for the following tasks:
 - Starting and stopping of resources
 - Moving of resources
 - Excluding resources on specific systems
 - Initiating SA OS/390 vote requests against resources
- ▶ To simulate a failure or an unplanned outage of resources, an external action must be taken, such as:
 - Kill a UNIX process ID
 - Cancel or stop an address space
 - Reset an LPAR
 - Stop a network adapter or power down a switch
 - Pull a cable

6.2.7 Verifications after the test

After each test, we first review the status of all the components using the same checklist as the one used before the test (see 6.2.4, “Verification of resource status” on page 137).

Then, depending on the type of scenario (usually in the case of a failure), we may do some additional verifications, such as:

- ▶ Looking at the SAP system log (SM21)
- ▶ Searching the SAP developer trace files for error messages
In which file an error is recorded may vary with the release of SAP. With SAP 4.6D, we exploit the following files:
 - dev_ms and dev_enqserv for errors regarding the message server and the enqueue server
 - dev_disp for errors regarding the connection to the message server
 - dev_w0 (for example) for errors regarding the connection to the enqueue server and the message server
- ▶ Displaying the status of internal and TCP/IP connections (SM59)
- ▶ Checking whether the workload we created is still running (SM66)
- ▶ Checking the number of lock entries in the enqueue table (SM12)
- ▶ Looking at the ICL message log ICL.<pid>.err1
- ▶ Displaying the DB2 threads using the DB2 command -DIS THREAD(*)

Important: A new trace file called enquelog has been introduced to log the activity of the enqueue server and the status of the replication.

Figure 6-11 is an extract of the new enqueue server log file. In our configuration, this file is located in the following directory: /usr/sap/RED/EM00/work/enquelog.

```
Start: Thu May 30 11:34:57 2002: enqueue server started
RepAct: Thu May 30 11:41:22 2002: replication activated
RepDea: Thu May 30 14:15:20 2002: replication deactivated
Stop: Thu May 30 14:15:36 2002: enqueue server stopped: normal shutdown
Start: Thu May 30 14:16:20 2002: enqueue server started
RepAct: Thu May 30 14:21:39 2002: replication activated
```

Figure 6-11 Extract of the new enqueue server log file

6.2.8 Analyze problems

If the results differ from the expected behavior, it is necessary to understand why. We put together some tips to help you with this complex troubleshooting phase in Chapter 7, “Problem determination methodology” on page 191.

6.3 Planned outage scenarios

In this section, we describe the planned outage scenarios that we chose to run at the ITSO in order to verify the SA OS/390 policy.

For each scenario, we specified the following:

- ▶ The purpose of the test
- ▶ The expected behavior
- ▶ The initial setup
- ▶ The phases of the execution
- ▶ The results we observed

In 6.2.4, “Verification of resource status” on page 137 we describe the verification tasks that we performed before and after each test to check the status of the SAP-related components. In this section, we do not repeat these steps. However, the description of each test may contain additional verification tasks that are specific to the scenario.

6.3.1 Stop and start of the entire SAP RED system

In this scenario, we wanted to test the normal stop and restart of the entire SAP RED system (including application servers, enqueue servers, database servers, etc.) using SA OS/390. We split this scenario into two parts: first the stop of the SAP system, and then the restart.

Table 6-2 summarizes the execution of the stop phase.

Table 6-2 Stop of the entire SAP system with SA OS/390

Purpose	Scope: The entire SAP RED system Action: Planned stop using SA OS/390
Expected behavior	All RED-related resources should come down properly. The NFS server, saprouter, and saposcol should stay up.
Setup	SC42 and SC04 must be up, including all required z/OS resources and SAP-related resources.

Execution	Issue a STOP request in SA OS/390 against the application group RED_SAPPLEX.
Results	All RED-related resources came down properly. The NFS server, saprouter, and saposcol stayed up.

Table 6-3 summarizes the execution of the start phase.

Table 6-3 Start of the entire SAP system with SA OS/390

Purpose	Scope: The entire SAP RED system Action: Planned start using SA OS/390
Expected behavior	All RED-related resources should come up properly.
Setup	SC42 and SC04 must be up, with all required z/OS resources, but all RED-related resources are stopped.
Execution	Kill the STOP request in SA OS/390 against the application group RED_SAPPLEX.
Results	All RED-related resources came up properly.

Figure 6-12 on page 149 shows the status of all the SAP-related resources before the test. All the local applications are in UP status. The NFS server and the enqueue server are running on SC04. The enqueue replication server is running on SC42.

S A P High Availability					
Local Applications		Moving Applications			
SC04	SC42	SC04	SC42		
-----	-----	-----	-----		
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA		
RED_DB2DBM1	RED_DB2DBM1				
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA		
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER		
RED_DB2SPAS	RED_DB2SPAS				
		RED_VIPA	RED_VIPA		
RED_RFC	RED_RFC	RED_ES	RED_ES		
REDICLI6	REDICLI6	RED_MS	RED_MS		
REDICLI7	REDICLI7	RED_GW	RED_GW		
REDICLI8	REDICLI8	RED_CO	RED_CO		
REDICLI9	REDICLI9	RED_SE	RED_SE		
		RED_ERS	RED_ERS		
APPSRV11	APPSRV10	APPSRV06	APPSRV06		
SAP_OSCOL	SAP_OSCOL	APPSRV07	APPSRV07		
		APPSRV08	APPSRV08		
				06/06/02 19:21	

Figure 6-12 Status of the SAP-related components before the test

To stop the entire SAP system, we issued a STOP request against the application group RED_SAPPLEX (option C); see Figure 6-13.

INGKYSTO					
Domain ID = SC04A		SA 05/390 - Command Dialogs		Line 1 of 1	
Operator ID = NETOP1		----- INGLIST -----		Date = 06/06/02	
CMD: A Update		Sysplex = WTSCPLX1		Time = 19:21:01	
G Members		I INGSCHED		/ scroll	
H DISPTRG		J INGGROUP			
C Stop					
D INGRELS					
E INGVOTE					
F INGINFO					
CMD Name		Type System		Compound	
Desired		Observed		Nature	
-----		-----		-----	
C	RED_SAPPLEX	APG	SATISFACTORY	AVAILABLE	BASIC

Figure 6-13 Stop of the entire SAP system

We wanted a *normal* stop of the SAP RED system. Thus, we stayed with the default type NORM; see Figure 6-14 on page 150.

Note: Because of our SA OS/390 definitions, only the monitor for the remote application server running on Linux stopped. The application server itself stayed idle until the system was up again, and then it reconnected.

If we wanted to stop the remote application server, we needed to issue a STOP request with the option FORCE on the application group RED_RASPLEX before stopping the group RED_SAPPLEX.

INGKYRUO		SA OS/390 - Command Dialogs	Page 1 of 2
Domain ID	= SC04A	----- INGREQ -----	Date = 06/06/02
Operator ID	= NETOP1		Time = 19:21:04
Resource	=> RED_SAPPLEX/APG	format: name/type/system	
System	=>	System name, domain ID or sysplex name	
Request	=> STOP	Request type (START, UP or STOP, DOWN)	
Type	=> NORM	Type of processing (NORM/IMMED/FORCE/user) or ?	
Scope	=> ALL	Request scope (ONLY/CHILDREN/ALL)	
Priority	=> LOW	Priority of request (FORCE/HIGH/LOW)	
Expire	=>	Expiration date(yyyy-mm-dd), time(hh:mm)	
Timeout	=> 0 / MSG	Interval in minutes / Option (MSG/CANCEL)	
AutoRemove	=>	Remove when (SYSGONE, UNKNOWN)	
Restart	=> NO	Restart resource after shutdown (YES/NO)	
Override	=> NO	(ALL/NO/TRG/FLG/DPY/STS/UOW/INIT)	
Verify	=> YES	Check affected resources (YES/NO/WTOR)	
Precheck	=> YES	Precheck for flags and passes (YES/NO)	
Appl Parms	=>		

Figure 6-14 Stop of the entire SAP system (continued)

Then we went back to our SDF customized panel to see the status of all the RED-related resources being dynamically updated. All the resources went from an UP status to a STOPPING status, and finally to an AUTODOWN status.

Figure 6-15 on page 151 shows the status of the SAP-related resources after stopping the SAP system. All RED-related resources came down properly. The NFS server, saposcol, and saprouter were still running.

Attention: The SA OS/390 resource APPSRV06 appears with an AUTODOWN status although the remote application server APPSRV06 is still running on Linux. Only the monitor has stopped.

S A P High Availability					
Local Applications		Moving Applications			
SC04	SC42	SC04	SC42		
-----		-----			
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA		
RED_DB2DBM1	RED_DB2DBM1				
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA		
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER		
RED_DB2SPAS	RED_DB2SPAS				
		RED_VIPA	RED_VIPA		
RED_RFC	RED_RFC	RED_ES	RED_ES		
REDICLI6	REDICLI6	RED_MS	RED_MS		
REDICLI7	REDICLI7	RED_GW	RED_GW		
REDICLI8	REDICLI8	RED_CO	RED_CO		
REDICLI9	REDICLI9	RED_SE	RED_SE		
		RED_ERS	RED_ERS		
APPSRV11	APPSRV10	APPSRV06	APPSRV06		
SAP_OSCOL	SAP_OSCOL	APPSRV07	APPSRV07		
		APPSRV08	APPSRV08		
06/06/02 19:25					

Figure 6-15 Status of the SAP-related components after stop

To restart the SAP system, we had to kill the remaining MakeUnavailable vote on the application group RED_SAPPLEX; see Figure 6-16.

```

INGKYRQO                      SA OS/390 - Command Dialogs          Line
Domain ID    = SC04A          ----- INGVOTE -----             Date = 06/06/02
Operator ID  = NETOP1         Sysplex = WTSCPLX1                  Time = 19:25:57

Resource    ==>  RED_SAPPLEX/APG
System      ==>                               System name, domain id or sysplex name

Cmd: C cancel request  K Kill request    S show request details
Cmd Action WIN Request/Vote Data
-----
K  STOP    Y  Request      : MakeUnAvailable
                  Created   : 2002-06-06 19:21:23
                  Originator : OPER_NETOP1(NETOP1)
                  Priority    : 01720000      Should Be Down - Operator
                  Status      : Winning/Satisfied

```

Figure 6-16 Restart of the entire SAP system (continued)

Figure 6-17 on page 152 shows the status of the SAP-related resources after the test. All the local applications are in UP status. The enqueue server is running on SC04. The enqueue replication server is running on SC42.

S A P High Availability					
Local Applications		Moving Applications			
SC04	SC42	SC04	SC42		
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA		
RED_DB2DBM1	RED_DB2DBM1				
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA		
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER		
RED_DB2SPAS	RED_DB2SPAS				
		RED_VIPA	RED_VIPA		
RED_RFC	RED_RFC	RED_ES	RED_ES		
REDICLI6	REDICLI6	RED_MS	RED_MS		
REDICLI7	REDICLI7	RED_GW	RED_GW		
REDICLI8	REDICLI8	RED_CO	RED_CO		
REDICLI9	REDICLI9	RED_SE	RED_SE		
		RED_ERS	RED_ERS		
APPSRV11	APPSRV10	APPSRV06	APPSRV06		
SAP_OSCOL	SAP_OSCOL	APPSRV07	APPSRV07		
		APPSRV08	APPSRV08		
				06/06/02 19:28	

Figure 6-17 Status of the SAP-related components after the test

6.3.2 Startup of all LPARs one after the other

In this scenario, we wanted to test the normal startup of the LPARs, one after the other. We split this scenario into two parts: the startup of the first LPAR (in our case SC42), and then the startup of the second LPAR (in our case SC04).

Table 6-4 summarizes the startup of the first LPAR.

Table 6-4 Startup of the first LPAR

Purpose	Scope: One LPAR Action: Planned startup of an LPAR while the other one is down
Expected behavior	The LPAR should come up with all required address spaces including all SAP-related resources: database server, ICL, application server, rfscoscol, and saposcol, plus NFS server and enqueue server, but not enqueue replication server.
Setup	Both LPARs must be down. An HMC is required.
Execution	IPL SC42 from Z02RA1, address=3833, parm=3800R2M1

Results	SC42 came up with all required address spaces including all SAP-related resources: database server, ICLL, application server, rfcoscol, and saposcol, plus NFS server and enqueue server, but not enqueue replication server.
----------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 6-5 summarizes the startup of the second LPAR.

Table 6-5 Startup of the second LPAR

Purpose	Scope: One LPAR Action: Planned startup of an LPAR while the other one is up
Expected behavior	The LPAR should come up with all required address spaces including all SAP-related resources: database server, ICLL, application server, rfcoscol, and saposcol, plus enqueue replication server.
Setup	The first LPAR must be up with all required z/OS resources and SAP-related resources: database server, ICLL, application server, rfcoscol, and saposcol, plus NFS server and enqueue server. The second LPAR must be down. An HMC is required.
Execution	IPL SC04 from Z02RA1, address=3833, parm=3800R2M1
Results	SC04 came up with all required address spaces including all SAP-related resources: database server, ICLL, application server, rfcoscol, and saposcol, plus enqueue replication server.

6.3.3 Shutdown and restart of an LPAR

In this scenario, we wanted to test the shutdown and restart of the LPAR where the enqueue server and the NFS server are running. We split this scenario into two parts: first the shutdown, and then the restart of the LPAR.

Table 6-6 summarizes the execution of the shutdown phase.

Table 6-6 Shutdown of the LPAR where the ES and the NFS server are running

Purpose	Scope: One LPAR Action: Planned shutdown of the LPAR where the enqueue server and the NFS server are running
----------------	-----------------------------------------------------------------------------------------------------------------

Expected behavior	<p>The NFS server should move to the other LPAR.</p> <p>The enqueue server should move to the other LPAR.</p> <p>The enqueue replication server should stop or move to another LPAR if more than two LPARs are available.</p> <p>The application server on the remaining LPAR should reconnect to the message server and enqueue server.</p> <p>The LPAR should come down properly to the point where we can enter the following command to remove the LPAR from the sysplex:</p> <pre>/V XCF,<sysname>,OFFLINE</pre>
Setup	<p>SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with:</p> <ul style="list-style-type: none"> ▶ The enqueue server running on SC04. ▶ The enqueue replication server running on SC42. ▶ The NFS server running on SC04.
Execution	<p>Move the SAP critical components running on SC04 to SC42 (NFS server, enqueue server, and saprouter).</p> <p>Stop the remaining SAP-related resources on SC04 (application server, rfcoscol, saposcol, ICLl servers, and database server).</p> <p>Issue a STOP request in SA OS/390 against the system group SC04 using the NetView command SHUTSYS ALL.</p>
Verifications	<p>Check that the application server APPSRV10 on SC42 reconnects to the message server and enqueue server.</p>
Results	<p>The NFS server moved from SC04 to SC42.</p> <p>The enqueue server moved from SC04 to SC42.</p> <p>The enqueue replication server stopped (the application group RED_ERSPLEX has a status INHIBITED).</p> <p>The application server APPSRV10 on SC42 reconnected to the message server and enqueue server.</p> <p>SC04 came down properly to the point where we can enter:</p> <pre>/V XCF,SC04,OFFLINE</pre>

Table 6-7 summarizes the execution of the restart phase.

Table 6-7 Restart of the LPAR where the ES and the NFS server are running

Purpose	<p>Scope: One LPAR</p> <p>Action: Restart after planned shutdown of the LPAR where the enqueue server and the NFS server are running (in our case SC04)</p>
Expected behavior	<p>SC04 should come up with all required address spaces including database server, ICLl, application server, rfcoscol, and saposcol.</p> <p>The enqueue server and the NFS server should stay on SC42.</p> <p>The enqueue replication server should restart to SC04.</p>


```

INGKYGRA          SA OS/390 - Command Dialogs
Domain ID   = SC04A   ----- INGGROUP -----   Date = 06/07/02
Operator ID = NETOP1   Sysplex = WTSCPLX1         Time = 19:38:32

Specify or revise the following data:

System   =>          System name, domain id or sysplex name

Action   =>  EXCLUDE   EXCLUDE-AVOID-INCLUDE or ACTIVATE-PACIFY or RESET

Group(s) =>  NFS_HAPLEX/APG RED_EMPLX/APG RED_ERSPLEX/APG
             RED_RASPLX/APG SAP_RTPLEX/APG

System(s)=>  SC04

```

Figure 6-19 Move of the critical SAP components

Note that we also excluded SC04 for the resource RED_ERSPLEX. If we had had a third system, the enqueue replication server would have moved to that system. In our configuration, the enqueue replication server stopped and the application group RED_ERSPLEX remained in an INHIBITED status; see Figure 6-20.

```

Domain ID   = SC04A   ----- INGLIST -----   Date = 06/07/02
Operator ID = NETOP1   Sysplex = WTSCPLX1         Time = 19:43:16
CMD: A Update   B Start   C Stop   D INGRELS   E INGVOTE   F INGINFO
      G Members  H DISPTRG I INGSCHED J INGGROUP   / scroll
CMD Name      Type System   Compound   Desired   Observed   Nature
-----
RED_ERSPLEX   APG          INHIBITED   AVAILABLE   SOFTDOWN   MOVE

```

Figure 6-20 Status of the application group RED_ERSPLEX after the move

Figure 6-21 shows the status of the SAP-related resources after moving the resources from SC04 to SC42.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
		RED_VIPA	RED_VIPA
RED_RFC	RED_RFC	RED_ES	RED_ES
REDICLI6	REDICLI6	RED_MS	RED_MS
REDICLI7	REDICLI7	RED_GW	RED_GW
REDICLI8	REDICLI8	RED_CO	RED_CO
REDICLI9	REDICLI9	RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10		
SAP_OSCOL	SAP_OSCOL	APPSRV06	APPSRV06
		APPSRV07	APPSRV07
		APPSRV08	APPSRV08

Figure 6-21 Status of the SAP-related components after the move

Now we stopped the SAP-related resources that were still running on SC04: the application server APPSRV11, the ICL1 servers, saposcol, rfcoscol, and the DB2 subsystem.

Because of all the dependencies defined in the SA OS/390 policy, issuing a STOP request against the application group RED_DB2GRP on SC04 not only stops the DB2 subsystem, but if the parameter scope is set to ALL (default value), it also stops all the children: the application server APPSRV11, the ICL servers, and rfcoscol. SA OS/390 lists all the resources affected by the STOP request and asks for confirmation; see Figure 6-22 on page 158 and Figure 6-23 on page 158.

```

INGKYRUO          SA OS/390 - Command Dialogs          Page 1 of 2
Domain ID = SC04A ----- INGREQ ----- Date = 06/07/02
Operator ID = NETOP1                                     Time = 19:40:58

Resource => RED_DB2GRP/APG/SC04          format: name/type/system
System   =>          System name, domain ID or sysplex name

Request  => STOP          Request type (START, UP or STOP, DOWN)
Type     => NORM          Type of processing (NORM/IMMED/FORCE/user) or ?
Scope    => ALL           Request scope (ONLY/CHILDREN/ALL)
Priority  => LOW           Priority of request (FORCE/HIGH/LOW)
Expire    =>              , Expiration date(yyyy-mm-dd), time(hh:mm)
Timeout  => 0 / MSG       Interval in minutes / Option (MSG/CANCEL)
AutoRemove =>            Remove when (SYSGONE, UNKNOWN)
Restart  => NO           Restart resource after shutdown (YES/NO)
Override  => NO           (ALL/NO/TRG/FLG/DPY/STS/UOW/INIT)
Verify    => YES          Check affected resources (YES/NO/WTOR)
Precheck  => YES          Precheck for flags and passes (YES/NO)
Appl Parms =>

```

Figure 6-22 Stop of the application group RED_DB2GRP

```

AOFKVFY1          SA OS/390 - Command Dialogs          Line 1 of 8
Domain ID = SC04A ----- INGREQ ----- Date = 06/07/02
Operator ID = NETOP1                                     Time = 19:41:07

Verify list of affected resources for request STOP

CMD: S show overrides T show trigger details V show votes
Cmd Name      Type System TRG SVP W Action Type Observed Stat
-----
APPSRV11     APL SC04      Y STOP NORM AVAILABLE
RED_DB2GRP    APG SC04      Y STOP NORM AVAILABLE
RED_RFC       APL SC04      Y STOP NORM AVAILABLE
REDICLI6      APL SC04      Y STOP NORM AVAILABLE
REDICLI7      APL SC04      Y STOP NORM AVAILABLE
REDICLI8      APL SC04      Y STOP NORM AVAILABLE
REDICLI9      APL SC04      Y STOP NORM AVAILABLE
RED_RASGRP    APG SC04      Y SOFTDOWN

```

Figure 6-23 Stop of the application group RED_DB2GRP (continued)

Then we issued a STOP request against the application group SAP_RTGRP on SC04. This stopped the saprouter.

Figure 6-24 on page 159 shows that there are no SAP-related resources left active on SC04.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
-----	-----	-----	-----
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
		RED_VIPA	RED_VIPA
RED_RFC	RED_RFC	RED_ES	RED_ES
REDICLI6	REDICLI6	RED_MS	RED_MS
REDICLI7	REDICLI7	RED_GW	RED_GW
REDICLI8	REDICLI8	RED_CO	RED_CO
REDICLI9	REDICLI9	RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10	APPSRV06	APPSRV06
SAP_OSCOL	SAP_OSCOL	APPSRV07	APPSRV07
		APPSRV08	APPSRV08
06/07/02 19:44			

Figure 6-24 Status of the SAP-related components after the stop

We could now take the system down using the NetView command SHUTSYS ALL; see Figure 6-25.

INGKYRUO		SA OS/390 - Command Dialogs		Page 1 of 2
Domain ID	= SC04A	----- INGREQ -----		Date = 06/07/02
Operator ID	= NETOP1			Time = 19:45:08
Resource	=> SC04/SYG/SC04	format: name/type/system		
System	=>	System name, domain ID or sysplex name		
Request	=> STOP	Request type (START, UP or STOP, DOWN)		
Type	=> NORM	Type of processing (NORM/IMMED/FORCE/user) or ?		
Scope	=> ALL	Request scope (ONLY/CHILDREN/ALL)		
Priority	=> LOW	Priority of request (FORCE/HIGH/LOW)		
Expire	=>	Expiration date(yyyy-mm-dd), time(hh:mm)		
Timeout	=> 0 / MSG	Interval in minutes / Option (MSG/CANCEL)		
AutoRemove	=>	Remove when (SYSGONE, UNKNOWN)		
Restart	=> NO	Restart resource after shutdown (YES/NO)		
Override	=> NO	(ALL/NO/TRG/FLG/DPY/STS/UOW/INIT)		
Verify	=> YES	Check affected resources (YES/NO/WTOR)		
Precheck	=> YES	Precheck for flags and passes (YES/NO)		
Appl Parns	=>			

Figure 6-25 Shutdown of the LPAR

SC04 came down to the point where we could enter the following MVS command to remove SC04 from the sysplex:

```
/V XCF,SC04,OFFLINE
```

Let's check that the application server APPSRV10 on SC42 reconnected to the message server and the enqueue server. Figure 6-26 is an extract of the developer trace dev_w0 for APPSRV10. It shows that APPSRV10 successfully reconnected to the message server and the enqueue server.

```
M Fri Jun 7 19:38:39 2002
M MBUF info for hooks: MS component DOWN
M ***LOG R0Z=> ThResetVBDISP, reset update dispatching info () Y./thxxvb.c 69
M *** ERROR => ThCheckReqInfo: message send/receive failed Y./thxxhead.c 13681"
M *** ERROR => ThMsOpcode: ThOpcodeToMsg failed (1) Y./thxxmsg.c 2769"
M ThVBHd1MsgDown: msg down
M ThIVBChangeState: update deactivated
M ***LOG R0R=> ThIVBChangeState, update deactivated () Y./thxxvb.c 9810"
M
M Fri Jun 7 19:39:09 2002
M MBUF info for hooks: MS component UP
M *** ERROR => ThSetEnqName: no enqueue server active Y./thxxtool.c 4163"
M ***LOG R1P=> ThSetEnqName, bad enq configuration () Y./thxxtool.c 4167"
S server '@>SSRV:wtsc04a_RED_11@<' vanished
S server '@>SSRV:vmlinux6_RED_00@<' vanished
M ThVBHd1MsgUp: msg up
M ThIVBChangeState: update activated
M ***LOG R0T=> ThIVBChangeState, update activated () Y./thxxvb.c 9796"
M
M Fri Jun 7 19:42:26 2002
M ***LOG Q0I=> NiPWrite: writev (140: EDC5140I Broken pipe.) Y./niuxi.c 1359"
M ENSA_DoRequest (): Reconnect
```

Figure 6-26 Extract of the developer trace dev_w0

Now we wanted SC04 up again.

We re-IPLed the LPAR. SA OS/390 was started automatically and restarted all the resources on the system, including the DB2 subsystem, the ICL servers, the application server APPSRV11, rfcoscol, and saposcol.

The enqueue replication server was not restarted because we still had the exclude of SC04 on the application group RED_ERSPLEX. To restart it, we removed this exclude (and all the outstanding excludes) using the NetView command INGGROUP; see Figure 6-27 on page 161.


```

INGKYGRA          SA OS/390 - Command Dialogs
Domain ID   = SC04A   ----- INGGROUP -----   Date = 06/07/02
Operator ID = NETOP1   Sysplex = WTSCPLX1         Time = 20:06:16

Specify or revise the following data:

System   =>          System name, domain id or sysplex name

Action   =>  INCLUDE  EXCLUDE-AVOID-INCLUDE or ACTIVATE-PACIFY or RESET

Group(s) =>  NFS_HAPLEX/APG RED_EMPLX/APG RED_ERSPLEX/APG
             RED_RASPLEX/APG SAP_RTPLEX/APG

System(s)=>  SC04

```

Figure 6-27 Restart of the enqueue replication server

As described in 6.2.4, “Verification of resource status” on page 137, we could also have used our “home-developed” REXX procedure SANCHK to remove the outstanding excludes.

The enqueue replication server started immediately on SC04.

Because we did not set any preferences in the policy to favor one LPAR or the other, the enqueue server and the NFS server stayed in place, on SC42.

We looked at the enqueue server log file `/usr/sap/RED/EM00/work/enquelog` to verify that the enqueue replication server reconnected to the enqueue server and that the replication was active. Figure 6-28 is the extract of this file corresponding to the time interval of our test.

```

RepDea: Fri Jun 7 19:38:40 2002: replication deactivated
Stop:   Fri Jun 7 19:38:43 2002: enqueue server stopped: normal shutdown
Start:  Fri Jun 7 19:38:58 2002: enqueue server started
RepAct: Fri Jun 7 20:06:26 2002: replication activated

```

Figure 6-28 Extract of the enqueue server log file

Figure 6-29 on page 162 shows the status of the SAP-related resources at the end of the test. All the SAP-related resources are in UP status. The NFS server and the enqueue server are running on SC42. The enqueue replication server is running on SC04.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
		RED_VIPA	RED_VIPA
RED_RFC	RED_RFC	RED_ES	RED_ES
REDICLI6	REDICLI6	RED_MS	RED_MS
REDICLI7	REDICLI7	RED_GW	RED_GW
REDICLI8	REDICLI8	RED_CO	RED_CO
REDICLI9	REDICLI9	RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10		
SAP_OSCOL	SAP_OSCOL	APPSRV06	APPSRV06
		APPSRV07	APPSRV07
		APPSRV08	APPSRV08
		06/07/02 20:05	

Figure 6-29 Status of the SAP-related components after the test

6.4 Unplanned outage scenarios

In this section, we describe the unplanned outage scenarios that we chose to run at the ITSO in order to verify the SA OS/390 policy.

For each scenario, we specified the following:

- ▶ The purpose of the test
- ▶ The expected behavior
- ▶ The initial setup
- ▶ The preparation for the test
- ▶ The phases of the execution
- ▶ The results we observed

In 6.2.4, “Verification of resource status” on page 137, we describe the verification tasks that we performed before and after each test to check the status of the SAP-related components. In this section, we do not repeat these steps. However, the description of each test may contain additional verification tasks that are specific to the scenario.

6.4.1 Failure of the enqueue server

In this scenario, we wanted to simulate the failure of the enqueue server and test the behavior of SA OS/390. We also wanted to measure the impact of the failure on the SAP workload.

Table 6-8 summarizes the execution of the test.

Table 6-8 Failure of the enqueue server

Purpose	Scope: Enqueue server Action: Unplanned outage
Expected behavior	SA OS/390 should show a PROBLEM/HARDDOWN status for the resource RED_ES and restart the enqueue instance (that is, all the members of the application group RED_EMGRP) on the LPAR where the enqueue replication server is running. The enqueue replication server should stop or move to another LPAR if more that two LPARs are available. The failure should be transparent to the SAP workload.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none">▶ The enqueue server running on SC42.▶ The enqueue replication server running on SC04.▶ The NFS server running on SC42.
Preparation	Log on to all the application servers. Create a workload on one application server (APPSRV11 on SC04). Create entries in the enqueue table.
Execution	Use the UNIX command <code>kill -9</code> to kill the enqueue server process externally (out of SA OS/390).
Verifications	Check that the workload is still running (SM66). Verify the number of entries in the enqueue table (SM12). Look for error messages in the enqueue log file, in the dev_enqserv file, in the developer traces dev_disp and dev_wx, and in the system log (SM21).
Results	SA OS/390 showed a PROBLEM/HARDDOWN status for RED_ES on SC42 and restarted the enqueue instance (that is, all the members of the application group RED_EMGRP) on SC04. The enqueue replication server stopped. The failure was transparent to the SAP workload.

Figure 6-30 on page 164 shows the status of the SAP-related resources before the test. All the SAP-related resources are in UP status. The NFS server and the enqueue server are running on SC42. The enqueue replication server is running on SC04.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
		RED_VIPA	RED_VIPA
RED_RFC	RED_RFC	RED_ES	RED_ES
REDICLI6	REDICLI6	RED_MS	RED_MS
REDICLI7	REDICLI7	RED_GW	RED_GW
REDICLI8	REDICLI8	RED_CO	RED_CO
REDICLI9	REDICLI9	RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10	APPSRV06	APPSRV06
SAP_OSCOL	SAP_OSCOL	APPSRV07	APPSRV07
		APPSRV08	APPSRV08
		06/12/02 16:09	

Figure 6-30 Status of the SAP-related components before the test

As described in 6.2.5, “Preparation for the test (unplanned outage only)” on page 142, we logged on to all the application servers, created a workload on APPSRV11 (five parallel tasks), and generated 10 lock entries in the enqueue table.

Then we simulated the failure: we killed the enqueue server process out of SA OS/390, using the UNIX command **kill -9 <pid>**, as shown on Figure 6-31.

SC42>ps -ef grep EM			
redadm	852529	17632351	- 15:23:30 ? 0:00 se.sapRED_EM00 -F pf=/usr/sap/RED/SYS/profile/RED_EM00
DFS	852860	17629600	- 16:10:01 tty0002 0:00 grep EM
redadm	853628	34408072	- 15:23:33 ? 0:00 co.sapRED_EM00 -F pf=/usr/sap/RED/SYS/profile/RED_EM00
redadm	853637	34408062	- 15:23:33 ? 0:06 es.sapRED_EM00 pf=/usr/sap/RED/SYS/profile/RED_EM00
redadm	855155	51186817	- 15:23:29 ? 0:00 gw.sapRED_EM00 pf=/usr/sap/RED/SYS/profile/RED_EM00
redadm	855172	34408031	- 15:23:30 ? 0:00 ms.sapRED_EM00 pf=/usr/sap/RED/SYS/profile/RED_EM00
SC42> kill -9 853637			

Figure 6-31 Simulation of the failure

Figure 6-32 shows that after the failure the resource RED_ES on SC42 is in the status PROBLEM/HARDDOWN.

INGKYSTO		SA OS/390 - Command Dialogs		Line 1 of 2	
Domain ID = SC04A		----- INGLIST -----		Date = 06/12/02	
Operator ID = NETOP1		Sysplex = WTSCPLX1		Time = 16:11:58	
CMD: A Update		B Start	C Stop	D INGRELS	E INGVOTE F INGINFO
G Members		H DISPTRG	I INGSCHED	J INGGROUP	/ scroll
CMD Name		Type System	Compound	Desired	Observed Nature
-----		-----		-----	
RED_ES		APL SC04	SATISFACTORY	AVAILABLE	AVAILABLE
RED_ES		APL SC42	PROBLEM	UNAVAILABLE	HARDDOWN

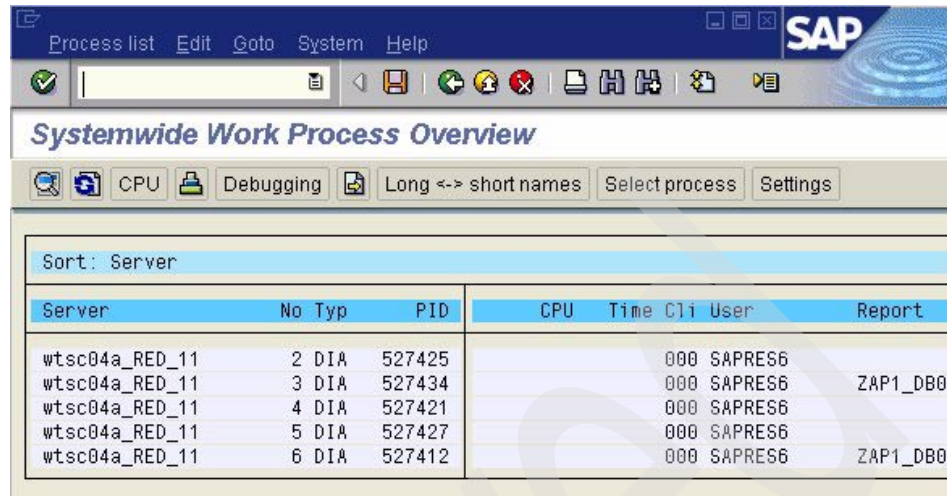
Figure 6-32 Status of RED_SE after the failover

Figure 6-33 shows the status of all the SAP-related resources after the failover. All the SAP-related resources are in UP status, except RED_ES on SC42. The NFS server is still running on SC42. The enqueue server is running on SC04. The enqueue replication server has stopped.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
-----		-----	
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
		RED_VIPA	RED_VIPA
RED_RFC	RED_RFC	RED_ES	RED_ES
REDICLI6	REDICLI6	RED_MS	RED_MS
REDICLI7	REDICLI7	RED_GW	RED_GW
REDICLI8	REDICLI8	RED_CO	RED_CO
REDICLI9	REDICLI9	RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10		
SAP_OSCOL	SAP_OSCOL	APPSRV06	APPSRV06
		APPSRV07	APPSRV07
		APPSRV08	APPSRV08
06/12/02 16:11			

Figure 6-33 Status of the SAP-related components after the failover

Using the transaction SM66, we verified that the five parallel tasks of our workload were still running after the failure; see Figure 6-34 on page 166.

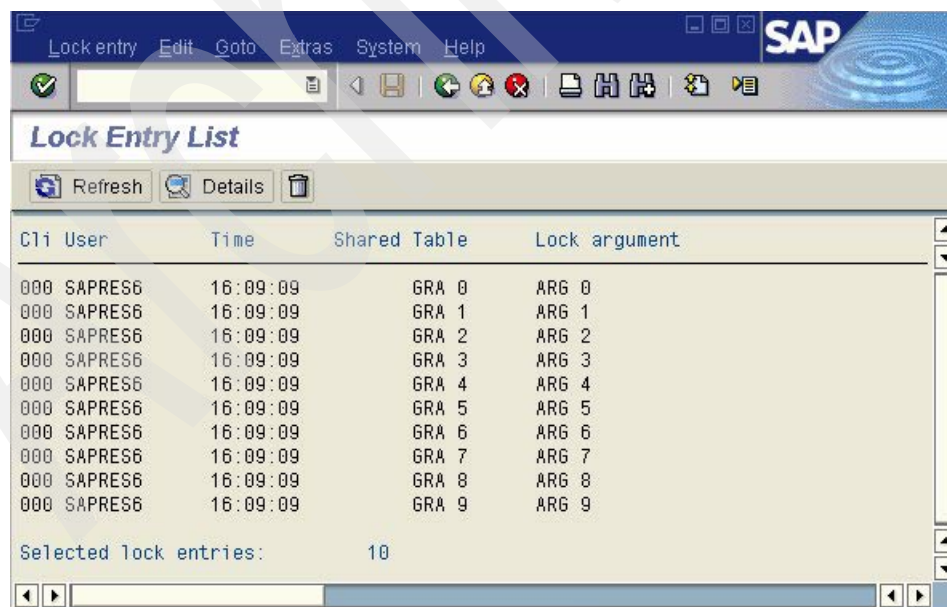


The screenshot shows the SAP 'Systemwide Work Process Overview' (SM66) window. The title bar includes 'Process list', 'Edit', 'Goto', 'System', and 'Help'. The menu bar contains 'Process list', 'Edit', 'Goto', 'System', and 'Help'. The toolbar has icons for CPU, Debugging, and Long <-> short names. The main area displays a table of work processes sorted by Server.

Server	No	Type	PID	CPU	Time	CLI	User	Report
wtsc04a_RED_11	2	DIA	527425			000	SAPRES6	
wtsc04a_RED_11	3	DIA	527434			000	SAPRES6	ZAP1_DB0
wtsc04a_RED_11	4	DIA	527421			000	SAPRES6	
wtsc04a_RED_11	5	DIA	527427			000	SAPRES6	
wtsc04a_RED_11	6	DIA	527412			000	SAPRES6	ZAP1_DB0

Figure 6-34 System-wide work process overview (SM66)

When the enqueue server restarts on SC04, it reads the enqueue replication table from shared memory and rebuilds the enqueue table. Using the transaction SM12, we verified that the 10 lock entries we had generated were still in the enqueue table; see Figure 6-35.



The screenshot shows the SAP 'Lock Entry List' (SM12) window. The title bar includes 'Lock entry', 'Edit', 'Goto', 'Extras', 'System', and 'Help'. The menu bar contains 'Lock entry', 'Edit', 'Goto', 'Extras', 'System', and 'Help'. The toolbar has icons for Refresh, Details, and a trash icon. The main area displays a table of lock entries.

CLI	User	Time	Shared	Table	Lock argument
000	SAPRES6	16:09:09	GRA	0	ARG 0
000	SAPRES6	16:09:09	GRA	1	ARG 1
000	SAPRES6	16:09:09	GRA	2	ARG 2
000	SAPRES6	16:09:09	GRA	3	ARG 3
000	SAPRES6	16:09:09	GRA	4	ARG 4
000	SAPRES6	16:09:09	GRA	5	ARG 5
000	SAPRES6	16:09:09	GRA	6	ARG 6
000	SAPRES6	16:09:09	GRA	7	ARG 7
000	SAPRES6	16:09:09	GRA	8	ARG 8
000	SAPRES6	16:09:09	GRA	9	ARG 9

Selected lock entries: 10

Figure 6-35 Lock entry list (SM12)

From the enqueue server log file `enquelog` (see Figure 6-36), we can see that the enqueue server restarted at 16:10:57 and that the enqueue replication server is not running (no message specifying that the replication is active).

```
Start: Wed Jun 12 16:10:57 2002: enqueue server started
```

Figure 6-36 Enqueue server log file `enquelog`

From the developer trace file `dev_disp` (see Figure 6-37), we can see that the dispatcher lost its connection with the message server at 16:10:05 and reconnected at 16:10:39.

```
Wed Jun 12 16:10:05 2002
*** ERROR => MsIReadFromHdl: NiRead (rc=-6) Y./msxxi.c 1434"
*** ERROR => NiBufCheckErr: Buffer 6 in stat err without errinfo Y./nibuf.c 1
*** ERROR => MsINiWrite: NiBufSend (rc=-1) Y./msxxi.c 2025"
*** ERROR => MsIDetach: MsINiWrite (rc=-100) Y./msxxi.c 957"
MBUF state OFF
MBUF component DOWN
*** ERROR => DpMsgProcess: MsReceive (-100) Y./dpxxdisp.c 1829"
*** ERROR => DpMsgProcess: disconnect from msg_server Y./dpxxdisp.c 1895"
*** ERROR => MsISnd2: not_attached Y./msxxi.c 1126"
***LOG QON=> DpRqNoWpHandle, MsSndName () Y./dpxxdisp.c 3286"

Wed Jun 12 16:10:19 2002
*** ERROR => MsIAttach: NiBufConnect to sapred/sapmsRED failed (rc=-10) Y./msxxi

Wed Jun 12 16:10:27 2002
*** ERROR => MsISnd2: not_attached Y./msxxi.c 1126"
***LOG QON=> DpRqNoWpHandle, MsSndName () Y./dpxxdisp.c 3286"

Wed Jun 12 16:10:39 2002
***LOG QOK=> DpMsAttach, mscon ( sapred) Y./dpxxdisp.c 7674"
MBUF set hwid_state to HWID_PENDING
DpMsgAdmin: Set release to 4640, patchlevel 0
MBUF state PREPARED
MBUF component UP
MBUF set hwid_state to SAP_0_K (X2102325001 )
DpMsgAdmin: Set patchno for this platform to 1145
Release check o.K.
MBUF state ACTIVE
```

Figure 6-37 Developer trace file `dev_disp`

We also looked at the developer trace file of one of the work processes on which the workload was running, for example `dev_w2` (see Figure 6-38). We can see that the work process lost its connection with the enqueue server at 16:10:04 and reconnected just after the enqueue server started (no more error messages after 16:10:57).

```

M Wed Jun 12 16:10:04 2002
M ***LOG QOI=> NiPWrite: writev (140: EDC5140I Broken pipe.) Y./niuxi.c 1359"
M ENSA_DoRequest (): Reconnect
M ***LOG QOI=> NiPConnect2: SO_ERROR (1128: EDC8128I Connection refused.) Y./ni
M EncNiConnect: unable to connect
M
M Wed Jun 12 16:10:05 2002
M EncNiConnect: unable to connect
M
M Wed Jun 12 16:10:07 2002
M EncNiConnect: unable to connect
M
M ....
M Wed Jun 12 16:10:57 2002
M EncNiConnect: unable to connect

```

Figure 6-38 Developer trace file dev_w2

Figure 6-39 shows the messages written in the SAP system log (SM21) during the interval of the test.

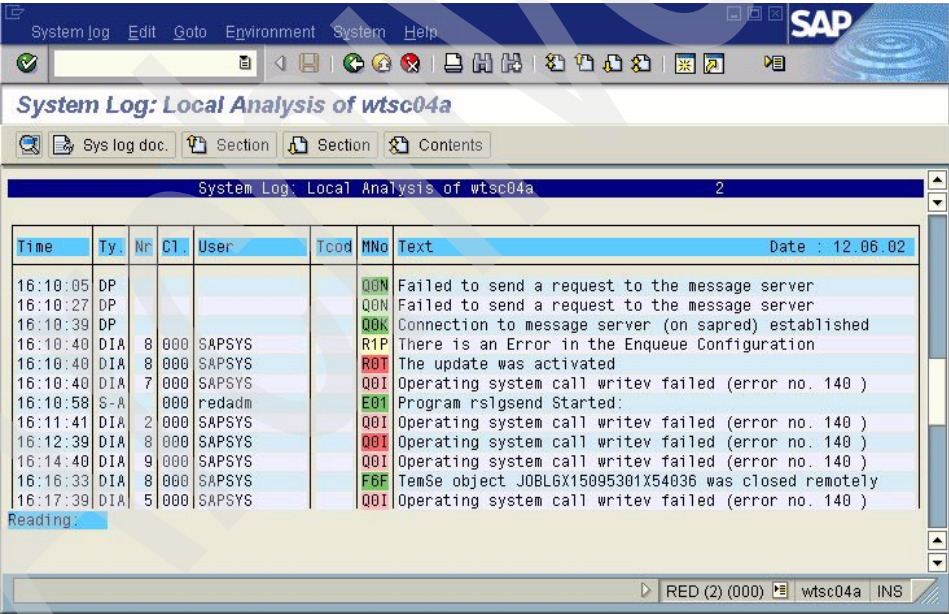


Figure 6-39 SAP system log (SM21)

Note that the system log shows a 6-minute interval before complete reconnection of the application server. This was due to a bug in TCP/IP, probably related to our multiple-stack environment. After we made a slight change to the TCP/IP configuration, the recovery time was reduced to less than a minute.

Because we had only two systems, the enqueue replication server is stopped and the application group RED_ERSPLEX remains in an INHIBITED status; see Figure 6-40.

Note that if we had had a third system, SA OS/390 would have restarted the enqueue replication server on that system.

INGKYSTO				SA OS/390 - Command Dialogs		Line 1 of 2	
Domain ID = SC04A		----- INGLIST -----		Date = 06/12/02			
Operator ID = NETOP1		Sysplex = WTSCPLX1		Time = 16:20:17			
CMD: A Update		B Start	C Stop	D INGRELS	E INGVOTE	F INGINFO	
G Members		H DISPTRG	I INGSCHED	J INGGROUP	/ scroll		
CMD Name		Type System	Compound	Desired	Observed	Nature	
-----		-----	-----	-----	-----	-----	
RED_ERS		APL SC04	SATISFACTORY	UNAVAILABLE	SOFTDOWN		
RED_ERS		APL SC42	INHIBITED	UNAVAILABLE	SOFTDOWN		

Figure 6-40 Status of RED_ES after the failure

Because we knew the source of the failure, we could use the NetView command SETSTATE to tell SA OS/390 that the resource RED_ES on SC42 should be in the AUTODOWN state (see Figure 6-41).

AOFKSAST		SA OS/390 - Command Dialogs	
Domain ID = SC04A		----- SETSTATE -----	Date = 06/12/02
Operator ID = NETOP1			Time = 16:20:52
Specify subsystem and new state:			
Subsystem	=>	RED_ES	Subsystem to be processed
System	=>	SC42	System name, domain ID, sysplex name or *ALL
State	=>	AUTODOWN	Put subsystem in the following state: (RESTART / AUTODOWN / CTLDOWN / UP / ENDED / MOVED)
Scope	=>	ONLY	Change state for: ONLY - only this subsystem CHILDREN - only dependant subsystems ALL - resource and dependants
Start	=>	NO	Start or restart subsystem now (YES/NO)
Override	=>	NO	Override(Start=YES) (NO/FLG/TRG/SVP/EVT/ALL)

Figure 6-41 Setting the status of RED_ES

As a result of this command, the resource RED_ES on SC24 is set to the status AUTODOWN, and the enqueue replication server immediately restarts on SC42.

Figure 6-42 shows the status of the SAP-related resources at the end of the test. All the SAP-related resources are in UP status. The NFS server and the enqueue replication server are running on SC42. The enqueue server is running on SC04.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
-----	-----	-----	-----
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
		RED_VIPA	RED_VIPA
RED_RFC	RED_RFC	RED_ES	RED_ES
REDICLI6	REDICLI6	RED_MS	RED_MS
REDICLI7	REDICLI7	RED_GW	RED_GW
REDICLI8	REDICLI8	RED_CO	RED_CO
REDICLI9	REDICLI9	RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10		
SAP_OSCOL	SAP_OSCOL	APPSRV06	APPSRV06
		APPSRV07	APPSRV07
		APPSRV08	APPSRV08
06/12/02 16:21			

Figure 6-42 Status of the SAP-related components after the test

6.4.2 Failure of the message server

In this scenario we wanted to simulate the failure of the message server and test the behavior of SA OS/390. We also wanted to measure the impact of the failure on the SAP workload.

Table 6-9 summarizes the execution of the test.

Table 6-9 Failure of the message server

Purpose	Scope: Message server Action: Unplanned outage
Expected behavior	SA OS/390 should try to restart the message server in place until the critical threshold is reached (5 failures in 10 minutes). If the critical threshold is reached, SA OS/390 should show a PROBLEM/HARDDOWN status for the resource RED_MS and the entire enqueue instance will move to the other system. The failure should be transparent to the SAP workload.

Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> ▶ The enqueue server running on SC42. ▶ The enqueue replication server running on SC04. ▶ The NFS server running on SC42.
Preparation	Log on to all the application servers. Create a workload on one application server (APPSRV11 on SC04). Create entries in the enqueue table.
Execution	Use the UNIX command <code>ki11 -9</code> to kill the message server process externally (out of SA OS/390).
Verifications	Check that the workload is still running (SM66). Verify the number of entries in the enqueue table (SM12). Look for error messages in the developer trace dev_disp and in the system log (SM21).
Results	SA OS/390 restarted the message server in place, on SC42. The failure was transparent to the SAP workload.

Figure 6-43 shows the status of the SAP-related resources before the test. All the SAP-related resources are in UP status. The NFS server and the enqueue server are running on SC42. The enqueue replication server is running on SC04.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
RED_RFC	RED_RFC	RED_VIPA	RED_VIPA
REDICLI6	REDICLI6	RED_ES	RED_ES
REDICLI7	REDICLI7	RED_MS	RED_MS
REDICLI8	REDICLI8	RED_GW	RED_GW
REDICLI9	REDICLI9	RED_CO	RED_CO
		RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10		
SAP_OSCOL	SAP_OSCOL	APPSRV06	APPSRV06
		APPSRV07	APPSRV07
		APPSRV08	APPSRV08

06/13/02 10:54

Figure 6-43 Status of the SAP-related components before the test

As described in 6.2.5, “Preparation for the test (unplanned outage only)” on page 142, we logged on to all the application servers, created a workload on APPSRV11 (5 parallel tasks), and generated 10 lock entries in the enqueue table.

Then we simulated the failure: we killed the message server process out of SA OS/390, using the UNIX command `kill -9 <pid>`, as shown on Figure 6-44.

```
SC42>ps -ef | grep ms.sapRED_EM00
redadm 34408866 854437 - 09:47:44 ? 0:00 ms.sapRED_EM00 pf=/usr
/sap/RED/SYS/profile/RED_EM00
DFS 854747 51186380 - 10:54:55 ttty0003 0:00 grep ms.sapRED_EM00
SC42>kill -9 34408866
```

Figure 6-44 Simulation of the failure

Because the critical threshold was not reached, SA OS/390 immediately restarted the message server in place, on SC42. Figure 6-45 shows the status of all the SAP-related resources after the failover. They are all in a UP status.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
-----		-----	
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
RED_RFC	RED_RFC	RED_VIPA	RED_VIPA
REDICLI6	REDICLI6	RED_ES	RED_ES
REDICLI7	REDICLI7	RED_MS	RED_MS
REDICLI8	REDICLI8	RED_GW	RED_GW
REDICLI9	REDICLI9	RED_CO	RED_CO
		RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10		
SAP_OSCOL	SAP_OSCOL	APPSRV06	APPSRV06
		APPSRV07	APPSRV07
		APPSRV08	APPSRV08
06/13/02 10:55			

Figure 6-45 Status of the SAP-related components after the failover

The failure was transparent: the workload was still running (SM66), and the lock entries that we generated were still in the enqueue table (SM12).

The dispatcher lost its connection with the message server at 10:55:02 and reconnected at 10:55:09; see Figure 6-46.

```

Thu Jun 13 10:55:02 2002
*** ERROR => MsIReadFromHdl: NiRead (rc=-6) Y./msxxi.c 1434"
*** ERROR => NiBufCheckErr: Buffer 5 in stat err without errinfo Y./nibuf.c 1
*** ERROR => MsINiWrite: NiBufSend (rc=-1) Y./msxxi.c 2025"
*** ERROR => MsIDetach: MsINiWrite (rc=-100) Y./msxxi.c 957"
MBUF state OFF
MBUF component DOWN
*** ERROR => DpMsgProcess: MsReceive (-100) Y./dpxxdisp.c 1829"
*** ERROR => DpMsgProcess: disconnect from msg_server Y./dpxxdisp.c 1895"
*** ERROR => DpRqCheck: T19 in stat TM_SLOT_FREE Y./dpxxdisp.c 4469"
***LOG QOG=> DpRqBadHandle, bad_req ( DIA) Y./dpxxdisp.c 3639"
*** ERROR => BAD REQUEST - Reason: DpRqCheck failed (line 4039): Y./dpxxdisp.c 3
-IN-- sender_id DISPATCHER      tid 19   wp_ca_blk -1   wp_id -1
-IN-- action   SEND_TO_WP      uid 37   appc_ca_blk -1   type DIA
-IN-- new_stat NO_CHANGE      mode 0   len -1   rq_id 5969
-IN-- req_info MS_ERROR
*** ERROR => MsISnd2: not_attached Y./msxxi.c 1126"
***LOG QON=> DpRqNoWpHandle, MsSndName () Y./dpxxdisp.c 3286"

Thu Jun 13 10:55:09 2002
***LOG QOK=> DpMsAttach, mscon ( sapred) Y./dpxxdisp.c 7674"
MBUF set hwid_state to HWID_PENDING
DpMsgAdmin: Set release to 4640, patchlevel 0
MBUF state PREPARED
MBUF component UP
MBUF set hwid_state to SAP_O_K (Q1913961624 )
DpMsgAdmin: Set patchno for this platform to 1145
Release check o.K.
MBUF state ACTIVE

```

Figure 6-46 Developer trace file dev_disp

Figure 6-47 shows the messages written in the SAP system log (SM21) during the interval of the test.

System Log: Local Analysis of wtsc04a							2		
Time	Ty	Nr	Cl	User	Tcod	MNo	Text		Date : 13.06.02
10:55:02	DP					QOG	Request (type DIA) cannot be processed		
10:55:02	DIA	6 000		SAPSYS		R02	The update dispatch info was reset		
10:55:02	DP					QON	Failed to send a request to the message server		
10:55:02	DIA	6 000		SAPSYS		R0R	The connection was de-activated after a DB error		
10:55:09	DP					QOK	Connection to message server (on sapred) established		
10:55:09	DIA	5 000		SAPSYS		R1P	There is an Error in the Enqueue Configuration		
10:55:09	DIA	5 000		SAPSYS		R0T	The update was activated		
Reading:									

Figure 6-47 SAP system log (SM21)

6.4.3 Failure of the ICLI server

In this scenario, we wanted to simulate the failure of the ICLI server and test the behavior of SA OS/390. We also wanted to measure the impact of the failure on the SAP workload.

Table 6-10 summarizes the execution of the test.

Table 6-10 Failure of the ICLI server

Purpose	Scope: ICLI server Action: Unplanned outage
Expected behavior	SA OS/390 should try to restart the ICLI server until the critical threshold is reached. When that happens, SA OS/390 should show a PROBLEM/HARDDOWN status and the ICLI server will not be restarted. Running transactions should be rolled back. Work processes should reconnect either to the same database server, or failover to the standby database server.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none">▶ The enqueue server running on SC04.▶ The enqueue replication server running on SC42.▶ The NFS server running on SC04.
Preparation	Log on to the remote application server. Create a workload on the remote application server.
Execution	Cancel the address space REDICLI6 on SC42.
Verifications	Check if the workload is still running (SM66). Look for error messages in the system log (SM21) and in the developer traces dev_wx. Use transaction DB2 and the DB2 command -DIS THREAD(*) to determine where the application server is connected.
Results	Because the critical threshold was not reached, SA OS/390 restarted the ICLI server REDICLI6 in place, on SC42. Running transactions were rolled back. Because the ICLI server was restarted before failover timeout detection, work processes could reconnect to the database server on SC42.

Figure 6-48 on page 175 shows the status of the SAP-related resources before the test. All SAP-related resources are in UP status. The NFS server and the enqueue server are running on SC42. The enqueue replication server is running on SC04.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
		RED_VIPA	RED_VIPA
RED_RFC	RED_RFC	RED_ES	RED_ES
REDICLI6	REDICLI6	RED_MS	RED_MS
REDICLI7	REDICLI7	RED_GW	RED_GW
REDICLI8	REDICLI8	RED_CO	RED_CO
REDICLI9	REDICLI9	RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10	APPSRV06	APPSRV06
SAP_OSCOL	SAP_OSCOL	APPSRV07	APPSRV07
		APPSRV08	APPSRV08
		06/14/02 09:09	

Figure 6-48 Status of the SAP-related components before the test

We logged on to the application server APPSRV06 running on VMLINUX6.

We displayed the current DB host using the SAP transaction DB2. (On the first panel of transaction DB2, we clicked **Installation parameters** → **Database analysis** → **Switch DB host**. We selected **Refresh** → **Execute**). Figure 6-49 shows that, before the failure, APPSRV06 is connected to wtsc42a, its primary DB host. The standby DB host is wtsc04a.

Settings:	
Primary DB host	wtsc42a
Standby DB host	wtsc04a
Present DB host	wtsc42a
Operation:	
Operation completed successfully.	
New DB host	wtsc42a

Figure 6-49 Display of the current DB host (DB2)

We started the workload on APPSRV06 (5 parallel tasks). Then we simulated the failure by cancelling the ICL server address space. Because the critical threshold was not reached, SA OS/390 immediately restarted the ICL server in place, on SC42; see Figure 6-50 on page 176.

```

09:09:36.64 SAPRES6 00000290 -R0 SC42,C REDICLI6
...
09:09:40.56 STC09771 00000090 $HASP395 REDICLI6 ENDED
...
09:09:41.42 AWRK0342 00000290 S REDICLI6
...
09:09:42.71 STC10710 00000090 $HASP373 REDICLI6 STARTED

```

Figure 6-50 Simulation of the failure

Figure 6-51 shows the messages written in the SAP system log (SM21) during the interval of the test.

System Log: Local Analysis of vmlinux6							
2							
Time	Ty	Nr	CL	User	Tcod	MNo	Text
09:09:40	DIA	3	000	SAPRES5		R68	Perform rollback
09:09:40	DIA	5	000	SAPRES5		BYM	SQL error 0 (possibly a network error); WP in reconnect status
09:09:40	DIA	5	000	SAPRES5		R68	Perform rollback
09:09:40	DIA	4	000	SAPRES5		BYM	SQL error 0 (possibly a network error); WP in reconnect status
09:09:40	DIA	4	000	SAPRES5		R68	Perform rollback
09:09:40	DIA	2	000	SAPRES5		BYM	SQL error 0 (possibly a network error); WP in reconnect status
09:09:40	DIA	2	000	SAPRES5		R68	Perform rollback
09:09:40	DIA	1	000	SAPRES5		BYM	SQL error 0 (possibly a network error); WP in reconnect status
09:09:40	DIA	1	000	SAPRES5		R68	Perform rollback
09:09:40	DIA	4	000	SAPRES5		AB0	Run-time error "DBIF_RSQSQL_ERROR" occurred
09:09:40	DIA	2	000	SAPRES5		AB0	Run-time error "DBIF_RSQSQL_ERROR" occurred
09:09:40	DIA	5	000	SAPRES5		AB0	Run-time error "DBIF_RSQSQL_ERROR" occurred
09:09:40	DIA	3	000	SAPRES5		AB0	Run-time error "DBIF_RSQSQL_ERROR" occurred
09:09:40	DIA	1	000	SAPRES5		AB0	Run-time error "DBIF_RSQSQL_ERROR" occurred
09:09:41	DIA	5	000	SAPRES5		R47	Delete session 001 after error 024
09:09:41	DIA	2	000	SAPRES5		R47	Delete session 001 after error 024
09:09:41	DIA	4	000	SAPRES5		R47	Delete session 001 after error 024
09:09:41	DIA	1	000	SAPRES5		R47	Delete session 001 after error 024
09:09:41	DIA	3	000	SAPRES5		R47	Delete session 001 after error 024
09:09:41	DIA	0				BV4	Work process is in reconnect status
09:09:41	DIA	0	000	SAPRES6	SM65	R47	Delete session 003 after error 024
09:09:48	DIA	8				BV4	Work process is in reconnect status
09:09:49	DIA	8				BYI	Work process has left reconnect status
09:09:56	DIA	0				BYI	Work process has left reconnect status
09:09:57	DIA	1				BYI	Work process has left reconnect status
09:10:23	DIA	0				BV4	Work process is in reconnect status
09:10:24	DIA	0				BYI	Work process has left reconnect status
09:10:29	DIA	8				BV4	Work process is in reconnect status
09:10:29	DIA	8				BYI	Work process has left reconnect status
09:10:44	DIA	1				BV4	Work process is in reconnect status
09:10:44	DIA	1				BYI	Work process has left reconnect status

Figure 6-51 SAP system log (SM21)

The five running transactions receive a DB2 SQL error 0 and are rolled back. The work processes are put in a reconnect status. The running sessions are lost and need to be restarted by the user. Within seconds, the work processes reestablish the connection and leave the reconnect status.

The transaction DB2 shows that the current DB host is still wtsc42a. We checked with the DB2 command -DIS THREAD(*) that all the threads were connected to SC42. Connection information for each work process can be found in the developer trace files, dev_wx.

Attention: During our test, we observed that the work processes could reconnect to the primary database server. This was because the ICLI server was restarted before failover time-out detection. However, especially in the case of a heavy workload, you could experience a failover to the standby database server.

6.4.4 Failure of the NFS server

In this scenario, we wanted to simulate the failure of the NFS server and test the behavior of SA OS/390. We also wanted to measure the impact of the failure on the SAP workload.

Table 6-11 summarizes the execution of the test.

Table 6-11 Failure of the NFS server

Purpose	Scope: NFS server Action: Unplanned outage
Expected behavior	SA OS/390 should restart the NFS server. Existing NFS mounts should be reestablished. The failure should be transparent to the SAP workload.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> ▶ The enqueue server running on SC42. ▶ The enqueue replication server running on SC04. ▶ The NFS server running on SC42.
Preparation	Log on to all the application servers. Create a workload on a remote application server (APPSRV06). Create entries in the enqueue table.
Execution	Cancel the address space MVSNFSSA on SC42.
Verifications	Check that the workload is still running (SM66). Verify the number of entries in the enqueue table (SM12). Check that the file systems are accessible (AL11). Look for error messages in the system log (SM21).
Results	SA OS/390 restarted the NFS server. Existing NFS mounts were reestablished. The failure was transparent to the SAP workload.

Figure 6-52 shows the status of the SAP-related resources before the test. All the SAP-related resources are in UP status. The NFS server and the enqueue server are running on SC42. The enqueue replication server is running on SC04.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
-----	-----	-----	-----
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
		RED_VIPA	RED_VIPA
RED_RFC	RED_RFC	RED_ES	RED_ES
REDICLI6	REDICLI6	RED_MS	RED_MS
REDICLI7	REDICLI7	RED_GW	RED_GW
REDICLI8	REDICLI8	RED_CO	RED_CO
REDICLI9	REDICLI9	RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10		
SAP_OSCOL	SAP_OSCOL	APPSRV06	APPSRV06
		APPSRV07	APPSRV07
		APPSRV08	APPSRV08

06/13/02 11:30

Figure 6-52 Status of the SAP-related components before the test

As described in 6.2.5, “Preparation for the test (unplanned outage only)” on page 142, we logged on to all the application servers, created a workload on the remote application server APPSRV06 (5 parallel tasks), and generated 10 lock entries in the enqueue table.

Then we simulated the failure by cancelling the address space of the NFS server on SC42 using the following command:

```
/C MVSNFSSA
```

Because, at the time of the test, the effective preference of SC04 was higher than that of SC42, SA OS/390 immediately restarted the NFS sever on SC04 (along with its VIPA) and put the resource MVSNFSSA on SC42 in a RESTART status; see Figure 6-53 on page 179.

AOFKSTA5		SA OS/390 - Command Dialogs		Line 1 of 2	
Domain ID = SC04A		----- DISPSTAT -----		Date = 06/13/02	
Operator ID = NETOP1				Time = 11:30:12	
A ingauto B setstate C ingreq-stop D thresholds E explain F info G tree					
H trigger I service J all children K children L all parents M parents					
CMD	RESOURCE	STATUS	SYSTEM	JOB NAME	A I S R D RS TYPE Activity

	MVSNFSSA	UP	SC04	MVSNFSSA	Y Y Y Y Y Y MVS --none--
	MVSNFSSA	RESTART	SC42	MVSNFSSA	Y Y Y Y Y Y MVS --none--

Figure 6-53 Status of the resource MVSNFSSA after the failover

Figure 6-54 shows the status of all the SAP-related resources after the failover. All the SAP-related resources are in UP status. The enqueue server is running on SC42. The NFS server and the enqueue replication server are running on SC04.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
-----		-----	
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
		RED_VIPA	RED_VIPA
RED_RFC	RED_RFC	RED_ES	RED_ES
REDICLI6	REDICLI6	RED_MS	RED_MS
REDICLI7	REDICLI7	RED_GW	RED_GW
REDICLI8	REDICLI8	RED_CO	RED_CO
REDICLI9	REDICLI9	RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10		
SAP_OSCOL	SAP_OSCOL	APPSRV06	APPSRV06
		APPSRV07	APPSRV07
		APPSRV08	APPSRV08
06/13/02 11:30			

Figure 6-54 Status of the SAP-related components after the failover

The failure is transparent: the workload is still running (SM66) and the lock entries that we generated are still in the enqueue table (SM12). All the file systems that are NFS-mounted on VMLINUX6 are accessible (AL11). No error messages are written to the SAP system log (SM21).

6.4.5 Failure of a TCP/IP stack

In this scenario, we wanted to simulate the failure of the TCP/IP stack on the system where the enqueue server and the NFS server are running, and test the behavior of SA OS/390. We also wanted to measure the impact of the failure on the SAP workload.

Attention: We believe that the fact that we were running with multiple TCP/IP stacks had an impact on the results we observed during our test. You may experience a different behavior in your environment.

Table 6-12 summarizes the execution of the test.

Table 6-12 *Failure of a TCP/IP stack*

Purpose	Scope: TCP/IP stack Action: Unplanned outage
Expected behavior	SA OS/390 should try to restart the TCP/IP stack until the critical threshold is reached. If the critical threshold is reached, SA OS/390 should show a PROBLEM/HARDDOWN status and the TCP/IP stack will not be restarted. The NFS server should fail and SA OS/390 should restart it. The enqueue instance should fail and SA OS/390 should restart it on the LPAR where the enqueue replication server is running. SA OS/390 should try to restart the enqueue replication server on a different LPAR. The application server running on the LPAR where the failure occurs should fail and SA OS/390 should restart it. For the remote application server connected to the database server running on the LPAR where the failure occurs, running transactions should be rolled back and work processes should reconnect either to the same database server, or failover to the standby database server. For the application server running on the other LPAR, the failure should be transparent.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none">▶ The enqueue server running on SC42.▶ The enqueue replication server running on SC04.▶ The NFS server running on SC42.
Preparation	Log on to all the application servers. Create a workload on APPSRV11 running on SC04 and on APPSRV06 running on VMLINUX6 and connected to SC42. Create entries in the enqueue table.
Execution	Cancel the address space TCPIPA on SC42.

Verifications	<p>Check if the workload is still running (SM66).</p> <p>Verify the number of entries in the enqueue table (SM12).</p> <p>Look for error messages in the enqueue log file, in the developer traces dev_disp and dev_wx, and in the system log (SM21).</p>
Results	<p>SA OS/390 could restart the TCP/IP stack on SC42.</p> <p>The NFS server failed and SA OS/390 restarted it on SC04.</p> <p>The enqueue instance failed and SA OS/390 restarted it on SC04.</p> <p>SA OS/390 tried to restart the enqueue replication server on SC42.</p> <p>But the resource RED_ES on SC42 was in a STUCK status because the USS process was hanging. After we manually cancelled the process, the enqueue replication server could start on SC42.</p> <p>APPSRV10 running on SC42 failed. SA OS/390 restarted it.</p> <p>For APPSRV06 running on VMLINUX6 and connected to the database server on SC42, running transactions were rolled back and, because the TCP/IP stack was restarted before failover time-out detection, work processes could reconnect to the database server on SC42.</p> <p>For APPSRV11 running on SC04, the failure was transparent.</p>

Figure 6-55 shows the status of the SAP-related resources before the test. All the SAP-related resources are in UP status. The NFS server and the enqueue server are running on SC42. The enqueue replication server is running on SC04.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
RED_RFC	RED_RFC	RED_VIPA	RED_VIPA
REDICLI6	REDICLI6	RED_ES	RED_ES
REDICLI7	REDICLI7	RED_MS	RED_MS
REDICLI8	REDICLI8	RED_GW	RED_GW
REDICLI9	REDICLI9	RED_CO	RED_CO
		RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10		
SAP_OSCOL	SAP_OSCOL	APPSRV06	APPSRV06
		APPSRV07	APPSRV07
		APPSRV08	APPSRV08
06/17/02 10:20			

Figure 6-55 Status of the SAP-related components before the test

As described in 6.2.5, “Preparation for the test (unplanned outage only)” on page 142, we logged on to all the application servers, created a workload on APPSRV11 (5 parallel tasks) as well as on APPSRV06 (5 parallel tasks), and generated 10 lock entries in the enqueue table.

We simulated the failure by stopping TCPIPA on SC42 using the following command:

```
/P TCPIPA
```

Because the critical threshold was not reached, SA OS/390 immediately restarted TCPIPA on SC42; see Figure 6-56.

```
10:20:26.37 SAPRES6 00000290 P TCPIPA
...
10:20:31.37 STC11046 00000090 $HASP395 TCPIPA ENDED
...
10:20:32.15 AWRK0942 00000290 S TCPIPA
...
10:20:32.76 STC11974 00000090 $HASP373 TCPIPA STARTED
```

Figure 6-56 Simulation of the failure

The failure of the TCP/IP stack led to the failure of the NFS server, the enqueue instance, the saprouter, and the application server APPSRV10 running on SC42. The ICL servers, however, stayed up and reconnected as soon as TCP/IP was back.

SA OS/390 immediately restarted the NFS server on SC04.

SA OS/390 restarted the enqueue instance on the LPAR where the enqueue replication server was running, that is, SC04. The enqueue replication server stopped and SA OS/390 tried to restart it on SC42.

During our test, although SA OS/390 had successfully restarted the enqueue instance on SC04, the resource RED_ES on SC42 remained in a STUCK status—the USS process was hanging and we had to cancel it using the following command:

```
/C REDADMES,A=3FE
```

As soon as the process was cancelled, the enqueue replication server started on SC42 and the replication was activated, as shown in Figure 6-57 on page 183.

```
RepDea: Mon Jun 17 10:20:27 2002: replication deactivated
Start: Mon Jun 17 10:21:37 2002: enqueue server started
RepAct: Mon Jun 17 10:33:12 2002: replication activated
```

Figure 6-57 Enqueue server log file enqueue.log

We believe that this problem has to do with the fact that we were running with multiple TCP/IP stacks. Instead of recovering manually, we could have added the CANCEL command in the SA OS/390 policy, as last shutdown command for the resource RED_ES.

SA OS/390 immediately restarted the saprouter on SC04.

The application server APPSRV10 running on SC42 went down and was immediately restarted by SA OS/390. All the sessions connected to this application server were, of course, lost and needed to be restarted.

The application server APPSRV06 running on VMLINUX6 lost the connection to the database server on SC42. The five running transactions received a DB2 SQL error 0 and were rolled back. The work processes were put in a reconnect status. The running sessions were lost and needed to be restarted by the users. Within seconds, the work processes reestablished the connection and left the reconnect status.

The transaction DB2 showed that the current DB host was still wtsc42a. We used the DB2 command -DIS THREAD(*) to check that all the threads are connected to SC42. Connection information for each work process can be found in the developer trace files dev_wx.

Attention: During our test, we observed that the work processes could reconnect to the primary database server. This was because the TCP/IP stack was restarted before failover time-out detection. However, especially in the case of a heavy workload, you could experience a failover to the standby database server.

For the application server APPSRV11 running on SC04, the failure is transparent—the workload is still running (SM66) and the lock entries that we generated are still in the enqueue table (SM12). The developer trace dev_disp shows that the dispatcher lost its connection with the message server at 10:20:27 and reconnected at 10:21:48; see Figure 6-58 on page 184.

```

Mon Jun 17 10:20:27 2002
***LOG QOI=> NiPRead: recv (1121: EDC8121I Connection reset.) Y./niuxi.c 1198"
*** ERROR => MsIReadFromHdl: NiRead (rc=-6) Y./msxxi.c 1434"
*** ERROR => NiBufCheckErr: Buffer 5 in stat err without errno Y./nibuf.c 1
*** ERROR => MsINiWrite: NiBufSend (rc=-1) Y./msxxi.c 2025"
*** ERROR => MsIDetach: MsINiWrite (rc=-100) Y./msxxi.c 957"
MBUF state OFF
MBUF component DOWN
*** ERROR => DpMsgProcess: MsReceive (-100) Y./dpxxdisp.c 1829"
*** ERROR => DpMsgProcess: disconnect from msg_server Y./dpxxdisp.c 1895"
*** ERROR => MsISnd2: not_attached Y./msxxi.c 1126"
***LOG QON=> DpRqNoWpHandle, MsSndName () Y./dpxxdisp.c 3286"
...
Mon Jun 17 10:21:48 2002
***LOG QOK=> DpMsAttach, mscon ( sapred) Y./dpxxdisp.c 7674"
MBUF set hwid_state to HWID_PENDING
MBUF state PREPARED
MBUF component UP
MBUF set hwid_state to SAP_0_K (X2102325001 )
DpMsgAdmin: Set patchno for this platform to 1145
Release check o.K.
MBUF state ACTIVE

```

Figure 6-58 Developer trace file dev_disp

The developer trace dev_w0 shows that the work process lost its connection with the enqueue server at 10:20:28 and reconnected as soon as the enqueue server was available (no more error messages after 10:21:36); see Figure 6-59.

```

M Mon Jun 17 10:20:28 2002
M ***LOG QOI=> NiPWrite: writev (140: EDC5140I Broken pipe.) Y./niuxi.c 1359"
M ENSA_DoRequest (): Reconnect
M
M Mon Jun 17 10:20:33 2002
M EncNiConnect: unable to connect
M
...
M Mon Jun 17 10:21:36 2002
M EncNiConnect: unable to connect

```

Figure 6-59 Developer trace file dev_w0

Figure 6-60 on page 185 shows the status of all the SAP-related resources after the failover. All the SAP-related resources are in UP status. The NFS server and the enqueue server are running on SC04. The enqueue replication server is running on SC42.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
-----		-----	
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
		RED_VIPA	RED_VIPA
RED_RFC	RED_RFC	RED_ES	RED_ES
REDICLI6	REDICLI6	RED_MS	RED_MS
REDICLI7	REDICLI7	RED_GW	RED_GW
REDICLI8	REDICLI8	RED_CO	RED_CO
REDICLI9	REDICLI9	RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10		
SAP_OSCOL	SAP_OSCOL	APPSRV06	APPSRV06
		APPSRV07	APPSRV07
		APPSRV08	APPSRV08
			06/17/02 10:21

Figure 6-60 Status of the SAP-related components after the failover

6.4.6 Failure of an LPAR

In this scenario, we wanted to simulate the failure of the LPAR where the enqueue server and the NFS server were running and test the behavior of SA OS/390. We also wanted to measure the impact of the failure on the SAP workload.

Table 6-13 summarizes the execution of the test.

Table 6-13 Failure of the LPAR where the ES and the NFS server are running

Purpose	Scope: One LPAR Action: Unplanned outage
Expected behavior	ARM should restart the failing DB2 subsystem on another LPAR with the option RESTART(LIGHT). The DB2 subsystem will go down after successful startup. SA OS/390 should restart the NFS server on another LPAR. SA OS/390 should restart the enqueue instance on the LPAR where the enqueue replication server is running. The enqueue replication server should stop or move to another LPAR if more than two LPARs are available.

Expected behavior (continued)	For the remote application server connected to the database server running on the failing LPAR, running transactions should be rolled back and work processes should failover to the standby database server. For the application server running on the other LPAR, the failure should be transparent.
Setup	SC04 and SC42 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> ▶ The enqueue server running on SC42. ▶ The enqueue replication server running on SC04. ▶ The NFS server running on SC42.
Preparation	Log on to all the application servers. Create a workload on APPSRV11 running on SC04 and on APPSRV06 running on VMLINUX6 and connected to the database server on SC42. Create entries in the enqueue table.
Execution	System reset at the HMC for SC42.
Verifications	Check if the workload is still running (SM66). Verify the number of entries in the enqueue table (SM12). Look for error messages in the enqueue log file, in the developer traces dev_disp and dev_wx, and in the system log (SM21).
Results	ARM restarted the failing DB2 subsystem D7X1 on SC04 with the option RESTART(LIGHT). The DB2 subsystem went down after successful startup. SA OS/390 restarted the NFS server on SC04. SA OS/390 restarted the enqueue instance on SC04. The enqueue replication server stopped. For APPSRV06 running on VMLINUX6 and connected to the database server on SC42, running transactions were rolled back and work processes reconnected to the standby database server on SC04. For APPSRV11 running on SC04, the failure was transparent.

Figure 6-61 on page 187 shows the status of the SAP-related resources before the test. All the SAP-related resources are in UP status. The NFS server and the enqueue server are running on SC42. The enqueue replication server is running on SC04.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1		
RED_DB2IRLM	RED_DB2IRLM	SAP_RTVIPA	SAP_RTVIPA
RED_DB2DIST	RED_DB2DIST	SAP_ROUTER	SAP_ROUTER
RED_DB2SPAS	RED_DB2SPAS		
		RED_VIPA	RED_VIPA
RED_RFC	RED_RFC	RED_ES	RED_ES
REDICLI6	REDICLI6	RED_MS	RED_MS
REDICLI7	REDICLI7	RED_GW	RED_GW
REDICLI8	REDICLI8	RED_CO	RED_CO
REDICLI9	REDICLI9	RED_SE	RED_SE
		RED_ERS	RED_ERS
APPSRV11	APPSRV10	APPSRV06	APPSRV06
SAP_OSCOL	SAP_OSCOL	APPSRV07	APPSRV07
		APPSRV08	APPSRV08
06/18/02 14:45			

Figure 6-61 Status of the SAP-related components before the test

As described in 6.2.5, “Preparation for the test (unplanned outage only)” on page 142, we logged on to all the application servers, created a workload on APPSRV11 (5 parallel tasks) as well as on APPSRV06 (5 parallel tasks), and generated 10 lock entries in the enqueue table.

We simulated the failure by doing a system reset at the HMC.

We used the NetView command INGLIST */*/SC42 to display the status of the resources on SC42. They all appeared with a status INHIBITED/SYSGONE. Figure 6-62 shows, as an example, the status of the application group RED_DB2GRP.

INGKYSTO		SA OS/390 - Command Dialogs		Line 1 of 8	
Domain ID = SC04A		----- INGLIST -----		Date = 06/18/02	
Operator ID = NETOP1		Sysplex = WTSCPLX1		Time = 14:50:00	
CMD: A Update	B Start	C Stop	D INGRELS	E INGVOTE	F INGINFO
G Members	H DISPTRG	I INGSCHED	J INGGROUP	/ scroll	
CMD Name	Type System	Compound	Desired	Observed	Nature
---	-----	-----	-----	-----	-----
RED_DB2GRP	APG SC42	INHIBITED	AVAILABLE	PROBLEM	BASIC

Figure 6-62 Status of the application group RED_DB2GRP

Automatic Restart Manager (ARM) restarted the DB2 subsystem D7X1 on SC04 with the option RESTART(LIGHT) in order to quickly release the retained locks. When the start-up was complete, D7X1 stopped.

SA OS/390 restarted the NFS server on SC04.

SA OS/390 restarted the enqueue instance on the LPAR where the enqueue replication server was running (SC04).

Because we had only two LPARs, the enqueue replication server stopped. If a third LPAR had been available, SA OS/390 would have restarted the enqueue replication server on that LPAR.

The application server APPSRV06 running on VMLINUX6 lost the connection to the database server on SC42. The five running transactions received a DB2 SQL error 0 and were rolled back. The work processes were put in a reconnect status. The running sessions were lost and needed to be restarted. The work processes did a failover to the standby database server, reestablished the connection and left the reconnect status.

The transaction DB2 showed that the current DB host was now wtsc04a, as shown in Figure 6-63. We also checked, with the DB2 command -DIS THREAD(*), that all the threads are connected to SC04. Connection information for each work process can be found in the developer trace files dev_wx.

Settings:	
Primary DB host	wtsc42a
Standby DB host	wtsc04a
Present DB host	wtsc04a
Operation:	
Operation completed successfully.	
New DB host	wtsc04a

Figure 6-63 Display of the current DB host (DB2)

For the application server APPSRV11 running on SC04, the failure is transparent—the workload is still running (SM66) and the lock entries that we generated are still in the enqueue table (SM12). The developer trace dev_disp shows that the dispatcher lost its connection with the message server at 14:52:46 and reconnected at 14:52:53; see Figure 6-64 on page 189.

```

Tue Jun 18 14:52:46 2002
***LOG QOI=> NiPRead: recv (1121: EDC8121I Connection reset.) Y./niuxi.c 1198"
*** ERROR => MsIReadFromHdl: NiRead (rc=-6) Y./msxxi.c 1434"
*** ERROR => NiBufCheckErr: Buffer 3 in stat err without errno Y./nibuf.c 1
*** ERROR => MsINiWrite: NiBufSend (rc=-1) Y./msxxi.c 2025"
*** ERROR => MsIDetach: MsINiWrite (rc=-100) Y./msxxi.c 957"
MBUF state OFF
MBUF component DOWN
...
Tue Jun 18 14:52:53 2002
***LOG QOK=> DpMsAttach, mscon ( sapred) Y./dpxxdisp.c 7674"
MBUF set hwid_state to HWID_PENDING
DpMsgAdmin: Set release to 4640, patchlevel 0
MBUF state PREPARED
MBUF component UP
MBUF set hwid_state to SAP_0_K (X2102325001 )
DpMsgAdmin: Set patchno for this platform to 1145
Release check o.K.
MBUF state ACTIVE

```

Figure 6-64 Developer trace file dev_disp

The developer trace dev_w3 shows that the work process reconnected with the enqueue server at 14:55:13; see Figure 6-65.

```

M Tue Jun 18 14:52:46 2002
M MBUF info for hooks: MS component DOWN
M ***LOG ROZ=> ThResetVBDISP, reset update dispatching info () Y./thxxvb.c 69
M *** ERROR => ThCheckReqInfo: message send/receive failed Y./thxxhead.c 13681"
M *** ERROR => ThMsOpcode: ThOpcodeToMsg failed (1) Y./thxxmsg.c 2769"
M ThVBHdlMsgDown: msg down
M ThIVBChangeState: update deactivated
M ***LOG ROR=> ThIVBChangeState, update deactivated () Y./thxxvb.c 9810"
M
M Tue Jun 18 14:52:53 2002
M MBUF info for hooks: MS component UP
M *** ERROR => ThSetEnqName: no enqueue server active Y./thxxtool.c 4163"
M ***LOG RIP=> ThSetEnqName, bad enq configuration () Y./thxxtool.c 4167"
S server '@>SSRV:wtsc42a_RED_100<' vanished
S server '@>SSRV:vmlinux6_RED_000<' vanished
M ThVBHdlMsgUp: msg up
M ThIVBChangeState: update activated
M ***LOG ROT=> ThIVBChangeState, update activated () Y./thxxvb.c 9796"
M
M Tue Jun 18 14:55:13 2002
M ***LOG QOI=> NiPRead: recv (1121: EDC8121I Connection reset.) Y./niuxi.c 1198
M ENSA_DoRequest (): Reconnect

```

Figure 6-65 Developer trace file dev_w3

Figure 6-66 shows the status of all the SAP-related resources after the failover. All the SAP-related resources running on SC04, including the NFS server and enqueue server, are in UP status. The enqueue replication server is stopped.

S A P High Availability			
Local Applications		Moving Applications	
SC04	SC42	SC04	SC42
RED_DB2MSTR	RED_DB2MSTR	MVSNFSSA	MVSNFSSA
RED_DB2DBM1	RED_DB2DBM1	SAP_RTVIPA	SAP_RTVIPA
RED_DB2IRLM	RED_DB2IRLM	SAP_ROUTER	SAP_ROUTER
RED_DB2DIST	RED_DB2DIST	RED_VIPA	RED_VIPA
RED_DB2SPAS	RED_DB2SPAS	RED_ES	RED_ES
RED_RFC	RED_RFC	RED_MS	RED_MS
REDICLI6	REDICLI6	RED_GW	RED_GW
REDICLI7	REDICLI7	RED_CO	RED_CO
REDICLI8	REDICLI8	RED_SE	RED_SE
REDICLI9	REDICLI9	RED_ERS	RED_ERS
APPSRV11	APPSRV10	APPSRV06	APPSRV06
SAP_OSCOL	SAP_OSCOL	APPSRV07	APPSRV07
		APPSRV08	APPSRV08
06/18/02 14:55			

Figure 6-66 Status of the SAP-related components after the failover

Problem determination methodology

In this chapter, we describe how to perform problem determination for System Automation for OS/390 (SA/390) and for each of the critical SAP components.

7.1 SA/390 problem determination

SAP HA is a complex environment, and in such an environment, problems can occur. In this chapter we direct you to areas where you can check for problems if you encounter various errors.

7.1.1 NetView netlog

All messages flowing to NetView are kept in two VSAM log files, NETLOGP (primary netlog), and NETLOGS (secondary netlog). These log files are used in a wraparound manner. Depending on their size, these log files typically keep from a few hours of data, up to several days of data.

To browse through the active log file, enter this command on the NetView NCCF command line:

```
BR NETLOGA
```

There is also a front-end panel for the netlog browse, which you call by entering this command on the NetView NCCF command line:

```
BLOG
```

BLOG allows for all kinds of filtering. For help information, enter the following command on the NetView NCCF command line:

```
HELP BLOG
```

To save the contents of the netlogs to a printer or a sequential file, you might want to use the procedure CNMPRT, which resides in PROCLIB.

7.1.2 z/OS SYSLOG

The z/OS system log, called the syslog, contains many more messages than the NetView netlog.

When you locate the time stamp of suspicious error messages in the netlog, it's a good idea to use this time stamp to check the z/OS syslog to find out what was *really* going on at that time.

The z/OS syslog is always saved and kept for a long time (usually, for years), and can be used for later problem determination and documentation.

7.1.3 Message Processing Facility

Some messages that show up in the z/OS syslog do not show up in the NetView netlog. This filtering is done in the Message Processing Facility (MPF) of z/OS, and it is often the reason for automation not functioning properly.

Many problems related to NetView automation routines are related to missing or wrong MPF definitions. This includes System Automation for OS/390, as it uses the NetView automation mechanism as its base.

The parameter member of the Message Processing Facility resides in SYS1.PARMLIB, member MPFLSTxx, where xx is a suffix chosen by your system programmer (the default is 00). Here is a sample MPF member fragment:

```
.  
. .  
.DEFAULT,SUP(YES),RETAIN(YES),AUTO(YES)  
BPXF024I  
.DEFAULT,SUP(YES),RETAIN(YES),AUTO(NO)  
IEF125I  
.  
.
```

In MPFLSTxx, three different filters can be set:

- ▶ SUP(YES/NO)
 - YES, to suppress messages from the system console.
 - NO, no change to the “normal” behavior.
- ▶ RETAIN(YES/NO)
 - YES, messages should be stored in the z/OS syslog.
 - NO, to prevent messages from being stored in the z/OS syslog. (This is very uncommon.)
- ▶ AUTO(YES/NO)
 - YES, to forward messages to an automation tool (in our case, NetView).
 - NO, to prevent forwarding messages to NetView. If a message is not automated in NetView for performance reasons, it’s a good idea to suppress forwarding.

7.1.4 Problem determination in System Automation for OS/390

Problem determination in System Automation for OS/390 really depends on the kind of error you encounter, but you should check these areas for indications:

- ▶ SDF or NMC

- DISPINFO
- INGINFO

SDF or NMC

The first indication of an unusual situation is often the dynamic display of SDF or NMC. This display shows the status of the resource in question. You can use the help function to learn more about the meaning of the status color of each resource. You can also use the EXPLAIN command on the NetView NCCF command line to see possible statuses and their meanings.

DISPINFO

The DISPINFO screen is not normally called directly from the command line (although it is possible), but rather out of the DISPSTAT panel. Thus you do not have to remember all the parameters; you can use convenient line commands instead. To get to the DISPINFO panel, enter: f (as shown in Figure 7-1).

AOFKSTA5				SA OS/390 - Command Dialogs				Line 21 of 45					
Domain ID = SC04A				----- DISPSTAT -----				Date = 06/21/02					
Operator ID = HTWANDR								Time = 10:10:28					
A ingauto		B setstate		C ingreq-stop		D thresholds		E explain		F info		G tree	
H trigger		I service		J all children		K children		L all parents		M parents			
CMD	RESOURCE	STATUS	SYSTEM	JOB NAME	A	I	S	R	D	RS	TYPE	Activity	

	RED_DB2SPAS	UP	SC04	D7X2SPAS	Y	Y	Y	Y	Y	Y	MVS	--none--	
	RED_ERS	AUTODOWN	SC04	REDADMER	Y	Y	Y	Y	Y	Y	MVS	--none--	
f	RED_ES	INACTIVE	SC04	REDADMES	Y	Y	Y	Y	Y	Y	MVS	--none--	
	RED_GW	INACTIVE	SC04	REDADMGW	Y	Y	Y	Y	Y	Y	MVS	--none--	
	RED_MS	INACTIVE	SC04	REDADMMS	Y	Y	Y	Y	Y	Y	MVS	--none--	
	RED_RFC	UP	SC04	REDADMR1	Y	Y	Y	Y	Y	Y	MVS	--none--	
	RED_SE	INACTIVE	SC04	REDADMSE	Y	Y	Y	Y	Y	Y	MVS	--none--	
	RED_VIPA	ENDED	SC04	TCPVIPA1	Y	Y	Y	Y	Y	Y	TRANS	--none--	

Figure 7-1 Calling the DISPINFO panel from DISPSTAT via line command f

Figure 7-2 on page 195 shows the DISPINFO panel.

```
AOFKINFO          SA OS/390 - Command Dialogs          Line 1    of 118
Domain ID   = SC04A   ----- DISPINFO -----         Date = 06/21/02
Operator ID = HTWANDR                                Time = 10:17:38

Subsystem ==> RED_ES      System ==> SC04      System name, domain ID
                                           or sysplex name
Subsystem   : RED_ES      on System : SC04

Description : SAP Enqueue Server

Class       : USS_APPL

Job Name    : REDADMES

Job Type    : MVS

Category    : USS

Current status : INACTIVE
Last Monitored : 10:15:46 on 06/21/02
Last Changed  : 15:33:54 on 06/20/02
Last Message  :
    AOF571I 15:33:54 : RED_ES SUBSYSTEM STATUS FOR JOB REDADMES IS
    INACTIVE - FAILED DURING START UP

Monitor      : AOFUXMON
Monitor Status : INACTIVE

      ---- shortened ----
```

Figure 7-2 Example of the DISPINFO panel

The DISPINFO panel provides useful information such as the following:

- ▶ Actual application status
- ▶ Date and time of last status change
- ▶ Start and stop commands
- ▶ Timeout values and threshold values for this application

INGINFO

The INGINFO screen is not normally called directly from the command line (although it is possible), but rather from the INGLIST panel. Thus you don't have to remember all the parameters; you can use convenient line commands instead (see Figure 7-3 on page 196).

INGKYSTO		SA OS/390 - Command Dialogs			Line 22 of 45	
Domain ID = SC04A		----- INGLIST -----			Date = 06/21/02	
Operator ID = HTWANDR		Sysplex = WTSCPLX1			Time = 10:24:51	
CMD: A Update		B Start	C Stop	D INGRELS	E INGVOTE	F INGINFO
G Members		H DISPTRG	I INGSCHED	J INGGROUP	/ scroll	
CMD Name	Type	System	Compound	Desired	Observed	Nature
-----	-----	-----	-----	-----	-----	-----
RED_ERS	APL	SC04	SATISFACTORY	UNAVAILABLE	SOFTDOWN	
f RED_ES	APL	SC04	PROBLEM	AVAILABLE	SOFTDOWN	
RED_GW	APL	SC04	PROBLEM	AVAILABLE	SOFTDOWN	
RED_MS	APL	SC04	PROBLEM	AVAILABLE	SOFTDOWN	
RED_RFC	APL	SC04	SATISFACTORY	AVAILABLE	AVAILABLE	
RED_SE	APL	SC04	PROBLEM	AVAILABLE	SOFTDOWN	
RED_VIPA	APL	SC04	SATISFACTORY	AVAILABLE	AVAILABLE	
REDICLI6	APL	SC04	SATISFACTORY	AVAILABLE	AVAILABLE	
REDICLI7	APL	SC04	SATISFACTORY	AVAILABLE	AVAILABLE	
REDICLI8	APL	SC04	SATISFACTORY	AVAILABLE	AVAILABLE	
REDICLI9	APL	SC04	SATISFACTORY	AVAILABLE	AVAILABLE	
RESOLVER	APL	SC04	SATISFACTORY	AVAILABLE	AVAILABLE	
RMF	APL	SC04	SATISFACTORY	AVAILABLE	AVAILABLE	

Figure 7-3 Calling the INGINFO panel from INGLIST via line command f

Figure 7-4 on page 197 shows an example of the INGINFO panel.

```

INGKYINO          SA OS/390 - Command Dialogs      Line 1    of 553
Domain ID   = SC04A   ----- INGINFO -----      Date = 06/21/02
Operator ID = HTWANDR      Sysplex = WTSCPLX1      Time = 10:25:32

Resource ==> RED_ES/APL/SC04          format: name/type/system
System    ==>          System name, domain ID or sysplex name

Resource      : RED_ES/APL/SC04
Category      : USS
Description    : SAP Enqueue Server

Status...
  Observed Status : SOFTDOWN
  Desired Status  : AVAILABLE
  Automation Status : PROBLEM
  Startable Status : YES
  Compound Status  : PROBLEM

Dependencies...
  PreStart        : Satisfied
  Start           : Satisfied
  PreStop         : Satisfied
  Stop            : Satisfied
  Startability    : Satisfied

--- shortened ---

```

Figure 7-4 Example of the INGINFO panel

In INGINFO you see information from the Automation Manager regarding the selected application, such as:

- ▶ The status, from the Automation Manager point of view
- ▶ The relationships of the application
- ▶ Open votes against the application
- ▶ The history of the last status changes to the resource

7.1.5 UNIX messages

By default, UNIX messages will not be sent to the z/OS syslog or to the NetView log. To send UNIX syslogd messages to the z/OS syslog, you must add an entry in the syslogd configuration file `/etc/syslog.conf`.

To forward all messages to the z/OS syslog, add the following entry:

```
*.* /dev/console
```

The UNIX messages will appear in the z/OS syslog with a BPXF024I message id. To send them further to NetView, you might have to modify MPF (see 7.1.3, “Message Processing Facility” on page 193).

7.1.6 If nothing happens

You may encounter a failure situation in which you enter a command to System Automation for OS/390 and nothing happens; there is no error message, and there are no “fancy lights” on SDF or NMC.

Typically this situation occurs because there is a lock in the system, which can have various causes. In this section, we describe these causes and show how you can determine where the problem lies:

- ▶ A pending vote
 - Use the INGVOTE command to look for open votes.
- ▶ Missing supporting applications
 - Check the relationships of the failing application. Are there any unresolved dependencies?
- ▶ Pending excludes or avoids against groups
 - Use the INGGROUP command or the SANCHK REXX to find excludes and avoids
- ▶ Auto flags in the System Automation for OS/390 agent
 - Enter: `DISPSTAT application name` and examine the automation flags. Using SA/390 2.1, they usually have to be switched on (Y).
- ▶ Disabled automation in the Automation Manager
 - Use the `a` line command on the INGLIST screen against the failing application, and check under action 3 for the automation manager auto flag.

7.1.7 When you are really lost

The last step before giving up and calling IBM support could be to do a cold start of the automation manager (HSAMPROC). A cold start will usually get rid of possible deadlocks, but note the following caveat.

Important: An automation manager cold start will also delete all dynamic overrides to thresholds, automation flags, schedules, preference values, and votes for all systems managed by the automation manager.

Usually the name of the automation managers started task is HSAMPROC, so after shutting down all automation managers (first the secondary, then the primary), enter the following start command at the z/OS system console:

```
s HSAMPROC,sub=mstr,type=cold
```

After the primary automation manager initializes, start the secondary automation managers.

7.1.8 Get help from the Web

A very useful table called “Tips for startup and shutdown problems” can be found at the following site:

<http://www.ibm.com/servers/eserver/zseries/software/sa/adds/hint02.html>

The table is part of the FAQ, hints & tips page. It is always worthwhile to browse through this table.

7.2 Where to check for application problems

This section describes where to look if System Automation indicates a problem with one of the defined UNIX applications, in particular with the SAP system.

- ▶ UNIX application cannot be started or stopped.
 - Check *.log files in the administrator’s home directory for error messages.

The name of the log file is specified in the start/stop/monitor command in System Automation, and it identifies resources and the system where the command has been executed. In our configuration, they are all located in the home directory /u/redadm.

The following command shows the log files in chronological order:

```
ls -rtl *.log
```

- Log file does not exist.

In this case, System Automation apparently either did not issue the USS command, or was unable to execute the command. You can do the following:

- Check the z/OS system log for messages (see “z/OS SYSLOG” on page 192).
- Check the USS system log (syslogd) for messages.
- Check the availability of file systems. Are the SAP global, profile, and exe directories accessible?
- Logon to USS and execute the command manually.

- For remote resources, the log files usually indicate the reason why System Automation failed to manage the resource. It may be that the remote resource is not truly unavailable—instead, remote monitoring, or remote execution, may be inhibited.
 - Check that the remote system is available.
 - Check that remote execution works.
 - Log on to the remote system and check the status.
- ▶ The SAP application server does not come up.
 - Check messages in the `startappsrv.*.log` file. This file contains the output of the startup command invoked by System Automation.

For debugging purposes, the script `startappsrv` contains an `env` command and an `ulimit` command at the beginning. This way, the process environment is made visible. You may add other commands as needed.

In our configuration, these files are located in the home directory `/u/redadm`.
 - Check messages in the `startsap_*.log` file. This file contains the output of the `startsap` command, which is invoked by `startappsrv`.
 - Check the SAP development traces in the work directory of the application server instance. List the files in chronological order to see which ones have been written last.

In our configuration, they are located in the directory `/usr/sap/RED/<appserver>/work`.
 - Check the home directory and the instance work directory for core files or CEEDUMP files indicating an abnormal termination of a UNIX process.

Such files are also written if a DLL was not found due to an incorrect `LIBPATH` environment variable, or a module could not be loaded because of a missing `STEPLIB` definition.
- ▶ SAP enqueue server, message server, gateway or syslog collector does not come up.

Problem determination in this case is similar to the application server case.

 - Check messages in the `startsap_EM00.*.log` file. This file contains the output of the startup command invoked by System Automation.
 - Check the SAP development traces in the work directory of the enqueue instance. List the files in chronological order to see which ones have been written last.

In our configuration, they are located in the directory `/usr/sap/RED/EM00/work`.

- For the enqueue server, browse the enqueolog file in the work directory. It shows when the enqueue server has been started and stopped, and whether the enqueue replication server is activated.
- A common startup problem of the syslog collector is that the global syslog file has become corrupted (this can happen, for example, if the file system is filled up).

The syslog file is located in the global directory and is named SLOGJ. Delete the file (the syslog collector will rebuild it automatically on its next startup).

In our configuration, it is located in the directory /usr/sap/RED/SYS/global.

- The application server do not connect to the message server or the enqueue server.

- Check the network and the routing; refer to “Checking the network” on page 201.
- Check that the enqueue server can be reached. For this purpose use the **ensmon** command:

```
ensmon -H <hostname> -I <enq_instance_number> 1
```

In our configuration, the command looks as follows:

```
ensmon -H sapred -I 00 1
```

The command writes further trace information into file dev_ensmon in the current directory. If ensmon fails on a remote system—but succeeds on the system where the enqueue server is running—the cause is probably a network problem.

7.3 Checking the network

Describing how to troubleshoot network problems could probably fill an entire volume. In this section, we mention just a few useful commands that you can use to verify the configuration and the connectivity between the systems. We also list commands to check the existence and location of dynamic VIPAs and the actual routing.

Note: You can issue these commands from different environments, such as: z/OS operator commands (OPER) format, TSO commands, and USS commands.

Checking the configuration

First, check the setup. The following command performs a basic consistency check:

TS0: HOMETEST

The following commands display the network configuration and attributes.

```
OPER: D TCPIP,,N,CONFIG
TSO:  NETSTAT CONFIG
USS:  netstat -f
```

The above command allows you to verify what you thought you had specified in the TCP/IP profile. In particular, check the following settings:

- ▶ FORWARDING YES
- ▶ IGREDIRECT 1
- ▶ SOURCEVIPA 1
- ▶ PATHMTUDSC 1

Note: If you use multiple TCP/IP stacks, you have to specify the name of the stack as the second parameter in the operator commands, as shown in the following example:

```
D TCPIP,TCPIPA,NE,CONFIG
```

Checking network devices

The following commands list the status of the interfaces:

```
OPER: D TCPIP,,N,DEV
TSO:  NETSTAT DEV
USS:  netstat -d
```

From the above commands, you can see the device status (for example, READY) and important facts such as whether it is configured as the PRI router (CFGROUTER), and whether it is currently used as the PRI router (ACTROUTER).

The next commands display the status of the interfaces, from an OSPF point of view:

```
OPER: D TCPIP,,OMPR,OSPF,IFS
```

Once you know the name of the interface from the second column of the display, you can gather more details by specifying the interface name as an additional parameter on this command:

```
OPER: D TCPIP,,OMPR,OSPF,IFS,NAME=<interface>
```

The DESIGNATED ROUTER for this interface is the router that makes all routing table changes for this interface (LAN) and broadcasts them. Of further interest are the STATE, the MAX PKT SIZE, and the number of NEIGHBORS and ADJACENCIES.

Dynamic VIPA

The following command displays the location and status of all VIPAs in the sysplex:

```
OPER: D TCPIP,,SYSPLEX,VIPADYN
```

In the USS environment, use the following command to display the list of home addresses (inclusive the VIPAs):

```
USS: netstat -h
```

or just the dynamic VIPAs on the system:

```
USS: netstat -v
```

Routing tables and OSPF

To display routing tables:

```
OPER: D TCPIP,,N,ROUTE
```

```
TSO: NETSTAT ROUTE
```

```
USS: netstat -r
```

To display gateways, you can use:

```
TSO: NETSTAT GATE
```

```
USS: netstat -g
```

To display OSPF tables:

```
OPER: D TCPIP,,OMPR,RTTABLE
```

Apart from the interface display that was previously explained, you may also want to see if OSPF is talking to its neighbors:

```
OPER: D TCPIP,,OMPR,OSPF,NBRS
```

You can even see statistical counters that show the quality of the conversations:

```
OPER: D TCPIP,,OMPR,OSPF,STATS
```

On AIX and Linux systems, the following command proved to be useful to watch the VIPA takeover among the z/OS systems. The -R option shows the current routing and indicates when the routing changes.

```
ping -R <hostname>
```

Checking active connections

To display all active IP connections on the system:

```
OPER: D TCPIP,,N,CONN
```

```
USS: netstat -c (or simply: netstat)
```

With this command you also see whether a static or dynamic VIPA is used as a source address or a target address, allowing you to easily verify that the SOURCEVIPAs option is effective (that is, for outgoing connections, the VIPA is used as a source address rather than the physical address of the network device).

7.4 Checking the status of the Shared HFS and of NFS

With the introduction of the Shared HFS, additional attributes have been added to the file system. They can be checked with the following command:

```
df -kv <filename>
```

Following is an example of the output and as you can see, the file system is currently owned by SC04 and is movable:

```
wtsc04a:/u/redadm (7)>df -kv /usr/sap/RED
Mounted on      Filesystem      Avail/Total      Files      Status
/sap/RED        (SAPRED.SHFS.SAPUSR)  2069924/2156400  4294966691  Available
HFS, Read/Write, Exported
File System Owner : SC04      Automove=Y      Client=N
Filetag : T=off  codeset=0
```

The following command allows the operator to check whether NFS clients have mounted a file system, (<MVS NFS> stands for the jobname of the NFS server):

```
F <MVS NFS>,LIST=MOUNTS
```

Consider the case where clients may not have done an explicit unmount (for example, if the connection was disrupted, or the client system was switched off). This usually does not impact the NFS server.

However, if an HFS dataset is unmounted and then remounted to the z/OS system, the NFS server does not allow NFS mounts to the newly available file system if any old NFS mounts are active.

The mount count is reset and unmounts are forced with the following command:

```
F <MVS NFS>,UNMOUNT='HFS/<mountpoint>'
```

Note: All clients will need to subsequently remount this NFS file system.

7.5 Checking the status of DB2 and SAP connections

In this section, we discuss basic techniques for identifying problems related to the SAP connections to DB2, or DB2 itself; we do not provide a comprehensive description of the general topic of problem determination. Additional problem

determination information can be found in *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966.

Check that DB2 is running

Use the SDSF DA command to show the status of DB2. (Prior to issuing this command, you can set your SDSF prefix to limit the display to DB2.)

For our configuration, we issued the following SDSF command:

```
pre d7x*
da
```

Figure 7-5 shows the results of these commands for our configuration. If the display doesn't show the DB2 systems running, then check the z/OS system log for messages (refer to "z/OS SYSLOG" on page 192).

Display	Filter	View	Print	Options	Help						

SDSF DA SC42	SC42	PAG	0 SIO	86 CPU	11/ 10	LINE 1-5 (5)					
COMMAND INPUT ==>						SCROLL ==> CSR					
PREFIX=D7X* DEST=(ALL) OWNER=* SORT=JOBNAME/A SYSNAME=SC42											
NP	JOBNAME	StepName	ProcStep	JobID	Owner	C Pos	DP	Real	Paging	SIO	
	D7X1DEM1	D7X1DBM1	IEFPROC	STC16226	STC	NS	FE	127T	0.00	0.00	
	D7X1DIST	D7X1DIST	IEFPROC	STC16229	STC	NS	FE	369	0.00	0.00	
	D7X1IRLM	D7X1IRLM		STC16222	STC	NS	FE	862	0.00	0.00	
	D7X1MSTR	D7X1MSTR	IEFPROC	STC16221	STC	NS	FE	773	0.00	0.18	
	D7X1SPAS	D7X1SPAS	IEFPROC	STC16231	STC	NS	FE	1136	0.00	0.00	

Figure 7-5 Results of SDSF DA command

Check the SAP database connections

- Use the DB2 Display Thread command to show the connections to DB2 from the SAP application server on USS, or the ICL server for remote application servers. The following is the command we used:

```
-D7X1 DISPLAY THREAD(*)
```

Figure 7-6 on page 207 shows the results of this command for our configuration. Notice that we have two application servers connected to DB2, wtsc42a (the USS application server), and vmlinux6 (the Linux application server).

For remote application servers (Linux6 is a remote application server in our configuration), you can check the ICL server message file for messages indicating that the ICL client on the application server tried to connect to the ICL server.

If these messages are present, then look for messages indicating why the ICLI server could not connect with DB2. In our configuration, the ICLI message files are located in the directory /usr/sap/RED/icli/icli6.

If there are no DB2 connections from the USS application servers, or the remote application servers haven't tried to connect to the ICLI servers, refer to "Where to check for application problems" on page 199.

```

Display Filter View Print Options Help
-----
SDSF OPERLOG DATE 07/02/2002 6 WTORS COLUMNS 52- 131
COMMAND INPUT ==> SCROLL ==> CSR
000290 -D7X1 DISPLAY THREAD(*)
000090 DSNV401I -D7X1 DISPLAY THREAD REPORT FOLLOWS -
000090 DSNV402I -D7X1 ACTIVE THREADS - 159
000090 NAME ST A REQ ID AUTHID PLAN ASID TOKEN
000090 RRSAP T 700 172021011001 REDADM CRED46C 0083 40
000090 V437-WORKSTATION= , USERID=*,
000090 APPLICATION NAME=wtsc42a
000090 RRSAP T 4302 172021011001 REDADM CRED46C 0083 41
000090 V437-WORKSTATION= # 6 h , USERID=*,
000090 APPLICATION NAME=wtsc42a
000090 RRSAP T 36 172021011001 REDADM SAPR346D 0070 38
000090 V437-WORKSTATION=6 00014 0000852704, USERID=*,
000090 APPLICATION NAME=wtsc42a
000090 RRSAP T 3067 172021011001 REDADM SAPR346D 007D 37
000090 V437-WORKSTATION=2 00013 0000852703, USERID=*,
000090 APPLICATION NAME=wtsc42a
.....
000090 RRSAP T 23362 192168050006 REDADM FOMEP46D 008A 14
000090 V437-WORKSTATION=1 00001 0000006748, USERID=10D6FA0000000006,
000090 APPLICATION NAME=vmlinux6
000090 RRSAP T 10362 192168050006 REDADM FOMEP46D 008A 15
000090 V437-WORKSTATION=1 00002 0000006749, USERID=10D78E2000000007,
000090 APPLICATION NAME=vmlinux6
000090 RRSAP T 220 192168050006 REDADM FOMEP46D 008A 17
000090 V437-WORKSTATION=1 00003 0000006750, USERID=10D7BF8000000008,
000090 APPLICATION NAME=vmlinux6
000090 RRSAP T 224 192168050006 REDADM FOMEP46D 008A 18
000090 V437-WORKSTATION=1 00005 0000006752, USERID=10D7D8300000000A,
000090 APPLICATION NAME=vmlinux6
.....
.... Shortened ....

```

Figure 7-6 Results of DB2 Display Thread command

Change management

This chapter discusses the procedures to update the SAP kernel and the ICL server and client in the environment presented in this redbook. It also discusses procedures on how to update DB2 and z/OS with minimal impact on the SAP application using z/OS Parallel Sysplex and DB2 data sharing.

8.1 Updating the SAP kernel

It's important for an SAP system that all application server instances use the same kernel level. For this reason, SAP has implemented a checking mechanism to ensure consistent kernels. In this section, we describe this mechanism in detail so you can understand why kernel updates have to follow a specific sequence.

- ▶ Each application server instance registers at the message server. The connection is established by the dispatcher. The dispatcher informs the message server—among other things—about the platform type (for example z/OS, Linux on zSeries, or AIX 64-bit) and its own patch level.
- ▶ The message server stores the patch level of the application server instance that connected first, but separately for each platform type. The value pairs—platform type plus corresponding patch level—are kept in memory as long as the message server is running. The values are *never* reset.
- ▶ When another instance registers later, the stored patch level for the corresponding platform is returned by the message server. If the dispatcher of that application server instance detects a mismatch, it stops.

Although SAP strongly recommends that the patch levels of all application server instances are identical, the checking mechanism enforces this rule only among instances of the same platform type. The reason for this is that sometimes a patch level is not available for all platforms.

While using the old central instance concept, this mechanism is very reasonable. The message server is started and stopped with the central instance. Therefore, the stored patch level is the one of the central instance.

However, with the new concept there are some implications. The application server instances might connect in arbitrary order. Furthermore, they are started and stopped independently of the message server. A new patch level for the instance (disp+work) usually does not affect the message server nor the enqueue server.

Beginning with kernel release 6.20, SAP introduces the *rolling kernel upgrade*. This concept handles the implications previously described, and is well suited for the HA environment.

8.1.1 Updating the SAP kernel (release 4.6)

As described in the preceding section, the first application server instance that connects to the message server defines the patch level. Application server

instances that connect afterwards must match the same patch level. The patch level is fixed until the message server is restarted.

Updating the dispatcher

If the dispatcher (disp+work) or one of its dynamically loaded modules (dbdb2slib.*, ibmiclic.*)¹ is to be updated, then perform the following steps. The sequence is applicable for UNIX systems including z/OS:

1. Save the old modules, which reside in the executable (exe/run) directory, and copy the new modules to this directory.
2. Stop all application server instances. Wait until all application servers are down.
3. Then stop and restart the message server.

In System Automation, this is accomplished by a STOP command with RESTART option set to YES.

4. Finally, start the application server instances again.

In System Automation, this is done by cancelling the STOP votes.

Note: On Windows, load modules cannot be replaced while they are in use. Therefore, first stop the application server instance before replacing the executables and dynamic load modules. On UNIX, shared objects (*.so) are locked and cannot be overwritten while they are in use. However, they can be renamed or moved to another directory.

Updating the enqueue server or replication server, or changing the size of the enqueue table

Updating components of the enqueue instance is quite easy, and it is transparent to the rest of the system.

If you want to update the *enqueue server* (enserver), simply let it fail over to the other system:

1. Save the old module which reside in the executable (exe/run) directory and copy the new module to this directory.
2. Move the enqueue instance to the system where the enqueue replication server is running.

In System Automation, this is accomplished by a STOP command on the enqueue server. Since the enqueue server is member of a MOVE group, it is

¹ With 6.20, the kernel is split into several load modules. The following dynamic load modules also belong to the kernel: dw_xml.*, dw_stl.*, dw_xtc.*.

automatically restarted on the system where the enqueue replication server is running on. Cancel the STOP vote afterwards.

If you want to increase the *size* of the enqueue table, you can take the same approach:

1. Modify the enqueue instance profile.
2. Move the enqueue instance to the system where the enqueue replication server is running.

If you want to update the *enqueue replication server* (enrepserver), perform these steps:

1. Save the old module, which resides in the executable (exe/run) directory, and copy the new module to this directory.
2. Then stop and restart the enqueue replication server.

In System Automation, this is accomplished by a STOP command with RESTART option set to YES. Afterwards, cancel this vote.

8.1.2 Rolling kernel upgrade

The concept described here is valid with SAP kernel release 6.20 and future releases.

The patch level consists of a major number and a minor number. If the major number is different, then the patch levels are incompatible. This happens if a communication protocol has changed or the ABAP runtime has a significant change. In this case, proceed as described in “Updating the dispatcher” on page 211”. However, this will be a rare situation.

If it is a compatible patch, you can perform a rolling kernel upgrade, as follows:

1. Save the old modules, which reside in the executable (exe/run) directory, and copy the new modules to this directory.
2. Stop and restart the application server instances. This can be done one after the other, or all at the same time.

The rolling kernel upgrade does *not* mean that the SAP system should run for a longer time with different patch levels. The rolling kernel upgrade should preferably be done while there are no active users or batch jobs. Stopping an instance implies that logged-in users have to reconnect and transactions which run on that instance are rolled back.

8.2 Updating the ICLI client and server

The ICLI client and server for a given SAP kernel release are characterized by two-level versioning:

1. Protocol version
2. Internal version

As long as the protocol version remains the same, then the ICLI server and client can be upgraded independently. The objective of the ICLI development team is to always keep the protocol compatible within one SAP kernel release. In fact, for 4.6D, the protocol version has never been changed.

The versions are displayed when invoking the ICLI server with the command line parameter -HELP. You also can find the protocol version of the current ICLI server in the log file (message ICLS1011I).

Note: The descriptions of rolling ICLI client and server upgrades given in the following sections apply as long as the protocol versions are the same.

8.2.1 Rolling upgrade of the ICLI client

The ICLI client can be downloaded and updated when the SAP kernel is updated (refer to “Updating the SAP kernel” on page 210).

If, for any reason, only the ICLI client is to be updated, proceed as follows. This procedure is valid for UNIX systems and has been tested on AIX 5.1 and on Linux for zSeries.

1. Save the old ICLI client.
2. Download the new client and adjust the permission bits, as described in *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966.
3. Choose *one* of the following options to restart the work processes.
 - a. Restart the application server instance.
 - b. Restart the work processes via SAP transaction SM50. In this transaction, the work processes are restarted by selecting **Process -> Cancel without core**.
 - c. Wait for the automatic restart of the work processes according to the respective SAP profile parameters. The following parameter settings mean that the work processes are started once a day (as suggested in SAP note 182207).

```
rdisp/wp_auto_restart = 86400  
rdisp/noptime = 87000
```

Each work process continues to run with the old ICLI client until it is restarted. When the work process restarts, it loads the new ICLI client.

8.2.2 Rolling upgrade of the ICLI server

The following method describes how to perform a rolling ICLI server update.

1. Apply the ICLI PTF.

As described in *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966, Chapter 8 “ICLI client and server”, the ICLI server load module resides in an APF-authorized library. By default it is SYS1.LINKLIB. The corresponding USS executable has the sticky bit turned on, which means that a load module with the same name is searched for in the usual z/OS library concatenation.

2. Perform a DB2 bind for the new packages *without binding the plan*.

For this step, take the sample bind job and remove the BIND PLAN step so that you only bind the packages. The package names have a unique name that corresponds to the ICLI version. New packages can be bound while the ICLI server has active database threads. Because the plan is not changed, it does not need to be bound again and, actually, it would not be possible to bind it while it is in use.

For more information on DB2 binding for the ICLI server can be found in *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966, Chapter 10.

3. Stop the ICLI servers and start them again.

Preferably perform this step when there are no long-running batch jobs, in order to avoid a lengthy rollback.

In a data sharing environment, we suggest that you first stop the ICLI server that is used as primary server. After restarting the primary server stop, and restart the standby server. Work processes that fail over when the primary server was stopped will automatically fail back when the standby server is stopped.

With System Automation, use the STOP command to stop the ICLI servers. In the sample policy, the STOP command uses the smooth stopping method that waits up to 60 seconds to allow a running transaction to complete its unit of work. Cancel the STOP vote and System Automation will automatically restart the ICLI servers.

8.2.3 Updating to an ICLI server with different protocol version

If the protocol version has changed, you can follow these steps to upgrade the ICLI client and server at the same time.

1. Apply the new ICLI PTF.
2. Perform a DB2 bind for the new packages *without binding the plan*.
3. Save the old ICLI client.
4. Download the new client and adjust the permission bits.

If you use more than one application server platform, repeat step 3 and 4 for each of them accordingly.

5. Stop the application servers.
6. Stop and restart the ICLI servers.
7. Restart the application servers.

8.3 Updating DB2 or z/OS

DB2 and z/OS can be updated by applying software maintenance, or upgrading to a newer release of the software. Applying software maintenance is done more often than upgrading the software. Software maintenance can be used to improve the performance of a software feature, or to fix software problems. In some special cases, new features can be added using software maintenance. SMP/E is the system tool used to apply, upgrade, track, and manage software for all z/OS products, including DB2 and z/OS.

At a very high level, SMP/E builds target executable libraries (loadlibs) while the software is executing from different executable libraries. In order to activate the latest changes, the software (z/OS or DB2) must be stopped and restarted using the updated loadlibs. For more detail on how to apply software maintenance using SMP/E, refer to the SMP/E User's Guide for the release of z/OS you are running.

Both DB2 and z/OS support downward compatibility. What that means is that you can run multiple software releases in a Parallel Sysplex data sharing environment. z/OS supports N +/- 3 releases. For example, OS/390 2.1, z/OS 1.1, and z/OS 1.2 can run in the same Parallel Sysplex.

DB2 supports N +/- 1 releases. For example, DB2 V6 and V7 can run in the same data sharing group. If both z/OS and DB2 need to be upgraded, the preferred sequence is to upgrade z/OS first, followed by DB2.

When z/OS Parallel Sysplex and DB2 data sharing are being used, the stopping and starting of z/OS and DB2 can be done without stopping the SAP system. This is accomplished by taking advantage of SAP's sysplex failover feature. The following steps should be used for each LPAR to be updated:

1. Build new DB2 loadlibs with the DB2 maintenance applied for each DB2 data sharing member.
A suggested name would be
<db2 member name>.SDSNLOAD.NEW
2. Stop all SAP R/3 batch processing on application servers connected to DB2 in this LPAR. Use "Opt Mode Switch" without any batch work processes to prevent new batch work from getting scheduled on this application server.
3. Activate SAP R/3 sysplex failover.
 - For SAP releases prior to 4.6, this is accomplished by stopping the primary ICL servers.
 - For SAP releases 4.6 and later, use SAP transaction DB2 for each application server connected to the DB2 data sharing members in the LPAR being updated.
4. Stop the DB2 data sharing members in the LPAR.
Issue a DB2 Display Thread command to ensure that there are no active connections to this DB2 member before issuing the Stop DB2 command.
5. Switch from current DB2 loadlibs to new DB2 loadlibs.
This can be accomplished by renaming the loadlibs as follows:
RENAME D7X1.SDSNLOAD to D7X1.SDSNLOAD.OLD
RENAME D7X1.SDSNLOAD.NEW to D7X1.SDSNLOAD
6. At this point, z/OS can be stopped and re-IPLed to activate z/OS updates.
7. Restart the DB2 data sharing members in the LPAR.
8. Restart any ICL servers that were previously stopped.
9. Resume normal configuration.
 - For SAP releases prior to 4.6, this is accomplished by stopping the standby ICL servers.
 - For SAP releases 4.6 and later, use SAP transaction DB2.
10. Restart all SAP R/3 batch processing on application servers connected to DB2 in this LPAR. Use "Opt Mode Switch" to add batch work processes.
11. Repeat step 1 through step 10 for each LPAR in the sysplex.

Network setup

This appendix contains the network configuration files for the SAP high availability solution configuration.

A.1 z/OS network settings

In the following section, we describe the network settings for z/OS.

A.1.1 z/OS network settings shared by both LPARs

We describe the following:

- ▶ z/OS UNIX System Services setup - BPXPRMxx
- ▶ TCP/IP started task
- ▶ Portmapper started task
- ▶ FTP daemon started task
- ▶ ICL server started task

z/OS UNIX System Services setup - BPXPRMxx

Following is a portion of the BPXPRMxx parmlib member used by both LPARs. It shows the network definitions for all TCP/IP stacks and NFS client definitions. It is executed on both LPARs.

```

/*****/
/*
/* The NETWORK statement defines which domain the specified
/* file system supports and some socket and port limits in that
/* domain by specifying:
/*
/*   o The address family type.
/*   o Its associated domain number.
/*   o The maximum number of sockets the address family will
/*     support.
/*   o The ports to be reserved for use with port zero,
/*     INADDR_ANY binds.
/* There must be a previous FILESYSTYPE statement that has a TYPE
/* operand that matches the TYPE operand on the NETWORK
/* statement.
/*
/* Currently, only two domains are supported:
/*   AF_UNIX, domain number 1, and entry point (BPXTUINT)
/*   AF_INET, domain number 2, and entry point (BPXTIINT) or
/*     entry point BPXTCINT, if Converged Sockets is used.
/*     (Converged Sockets is intended to be used only if
/*     multiple AF_INET physical file systems (such as 2
/*     TCP/IP's) are to be active at one time. There is a
/*     performance degradation if using Converged Sockets
/*     with a single AF_INET physical file system.)
/*
/* Port reservation information for port zero, INADDR_ANY binds
/* is only required for the AF_INET domain in a Common INET
/* configuration. It is specified with the INADDRANYPORT and
/* INADDRANYCOUNT parameters. If these parameters are omitted,
/* then no ports are reserved.
/*
/* INADDRANYPORT specifies the starting port number to be
/* reserved for use by OpenEdition applications that issue
/* port zero, INADDR_ANY binds. INADDRANYCOUNT specifies how
/* many ports to reserve.
/*
/* If you are running a Common INET configuration and you
/* specify the INADDRANYPORT and INADDRANYCOUNT parameters then
/* you must specify the same values to each transport provider
/* that is specified with the SUBFILESYSTYPE statement. Refer
/* to the documentation for that transport provider to determine
/* how the port reservation information is specified.
/*
/* In this example --
/*
/* For TYPE(UDS):
/*   o ENTRYPOINT is BPXTUINT,
/*   o DOMAINNAME is 'AF_UNIX'.
/*   o DOMAINNUMBER is 1.

```

```

/*      o MAXSOCKETS is 64.                                */
/*      o The TYPE of the file system is 'UDS'.             */
/*      o No port reservations are required for AF_UNIX.    */
/*                                                         */
/* For TYPE(INET):                                          */
/*      o ENTRYPOINT is BPXTIINT,                           */
/*      o DOMAINNAME is 'AF_INET'.                          */
/*      o DOMAINNUMBER is 2.                                */
/*      o MAXSOCKETS is 64.                                  */
/*      o The TYPE of the file system is 'INET'             */
/*      o No port reservations are required.                */
/*                                                         */
/*                                                         */
/* Notes:                                                  */
/*                                                         */
/* 1. The name specified for DOMAINNAME is the name that will */
/*    appear in messages referring to this address family. The */
/*    name specified can be any name up to 16 bytes in length. */
/* 2. The value specified for DOMAINNUMBER is what tells OMVS */
/*    that the address family being represented is AF_UNIX. For */
/*    a list of valid values for this operand, refer to BPXYSOCK. */
/* 3. MAXSOCKETS refers to the maximum number of sockets that can */
/*    be active at one time. Refer to the proper manual for */
/*    information on the maximum value that this operand can have. */
/* 4. The name specified for the TYPE operand must match the name */
/*    of a previous FILESYSTYPE statement.                  */
/*                                                         */
/*                                                         */
/*                                                         */
/*****/
FILESYSTYPE TYPE(UDS) ENTRYPOINT(BPXTUINT)
NETWORK DOMAINNAME(AF_UNIX)
        DOMAINNUMBER(1)
        MAXSOCKETS(1000)
        TYPE(UDS)

FILESYSTYPE TYPE(CINET)
        ENTRYPOINT(BPXTICINT)
NETWORK DOMAINNAME(AF_INET)
        DOMAINNUMBER(2)
        MAXSOCKETS(25000)
        TYPE(CINET)
        INADDRANYPORT(10000)
        INADDRANYCOUNT(2000)

SUBFILESYSTYPE NAME(TCPIPOE)
        TYPE(CINET)
        ENTRYPOINT(EZBPFINI)
        DEFAULT

```

```

SUBFILESYSTYPE NAME(TCIPMVS)
    TYPE(CINET)
    ENTRYPOINT(EZBPFINI)

```

```

SUBFILESYSTYPE NAME(TCPIP)
    TYPE(CINET)
    ENTRYPOINT(EZBPFINI)

```

```

SUBFILESYSTYPE NAME(TCPIPA)
    TYPE(CINET)
    ENTRYPOINT(EZBPFINI)

```

```

SUBFILESYSTYPE NAME(TCPIPB)
    TYPE(CINET)
    ENTRYPOINT(EZBPFINI)

```

```

SUBFILESYSTYPE NAME(TCPIPC)
    TYPE(CINET)
    ENTRYPOINT(EZBPFINI)

```

```

FILESYSTYPE TYPE(NFS) ENTRYPOINT(GFSCINIT)
    ASNAME(NFSCINT)
    PARM ('DISABLELLA(Y)')

```

```

/*****
/*
/* The SUBFILESYSTYPE statement specifies a socket file system
/* that the Open MVS Converged Sockets physical file system is
/* to start. This is used if more than one AF_INET physical
/* file system (TCP/IP) is active at the same time.
/*
/* The SUBFILESYSTYPE statement is associated with the
/* FILESYSTYPE statement that describes the Open MVS Converged
/* Sockets physical file system by matching the value specified
/* in the TYPE operand.
/*
/* In this sample:
/*   o 'INET' is the TYPE specified for this file system
/*   o The ENTRYPOINT 'BPXTCINT' is the name of the load module
/*     to be ATTACHed during initialization of the Converged
/*     Sockets physical file system.
/*   o The ENTRYPOINT 'BPXTIINT' is the name of the load module
/*     to be ATTACHed during initialization of the AF_INET
/*     Sockets physical file system.
/*   o PARM is not specified.
/*
/* Notes:
/*
*/

```

```

/* 1. NAME, TYPE and ENTRYPOINT are required parameters. */
/* 2. NAME can be up to 8 characters. It specifies the */
/* name by which this file system will be known to the */
/* Converged Sockets physical file system. In the case of */
/* TCP/IP, this is the procname. */
/* 3. TYPE can be up to 8 characters. It specifies the name of */
/* the Converged Sockets physical file system type identified */
/* in a FILESYSTYPE statement TYPE parameter. */
/* 4. ENTRYPOINT can be up to 8 characters. It specifies the */
/* name of of the load module containing the entry point into */
/* the file system type. */
/* 5. There can be multiple SUBFILESYSTYPE statements. */
/* 6. PARM can be up to 1024 characters. */
/* It must be entered as a quoted string. It can be entered */
/* in mixed case, as required by the physical file system, */
/* e.g. PARM ('/u'). */
/* A null PARM can be omitted, or optionally be specified */
/* as PARM(' '). */
/* 6. DEFAULT has no parameters. */
/* This operand is only useful if there are more than one */
/* SUBFILESYSTYPE statement. The sockets physical file system */
/* designated as the DEFAULT will be used by the Converged */
/* Sockets physical file system to set default routes and */
/* will be queried for host ID information. */
/* 7. The specific parameters and values for the parameters are */
/* file system dependent. Refer to the installation guide for */
/* file system that is to be started. */
/*
/*****

```

TCPIP started task

This procedure is used by both LPARs

```

//TCPIPA  PROC P1='CTRACE(CTIEZB00)',TCPprof=TCPprof,TCPDATA=TCPDATA
/*
//TCPIP  EXEC PGM=EZBTCPIP,REGION=OM,TIME=1440,PARM=&P1
//STEPLIB DD DSN=TCPIP.SEZATCP,DISP=SHR
/*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=VB,LRECL=137,BLKSIZE=0)
//SYSERR DD SYSOUT=*,DCB=(RECFM=VB,LRECL=137,BLKSIZE=0)
//SYSERROR DD SYSOUT=*
//CEEDUMP DD SYSOUT=*,DCB=(RECFM=VB,LRECL=137,BLKSIZE=0)
//PROFILE DD DSN=TCPIPA.&SYSNAME..TCPparms(&TCPprof.),
// DISP=SHR,FREE=CLOSE
//SYSTCPD DD DSN=TCPIPA.&SYSNAME..TCPparms(&TCPDATA.),DISP=SHR
//SYSABEND DD SYSOUT=*

```

Portmapper started task

This procedure is used by both LPARs.

```
//PORTMAP PROC P1='POSIX(ON) ALL31(ON)',
//          P2='ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIPA")'
//PMAP      EXEC PGM=OPORTMAP,REGION=OM,TIME=NOLIMIT,
//          PARM=('&P1 &P2 / ')
//STDOUT    DD PATH='/tmp/pmapa-stdout',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=SIRWXU
//STDERR    DD PATH='/tmp/pmapa-stderr',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=SIRWXU
//SYSOUT    DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

FTP daemon started task

This procedure is used by both LPARs.

```
//FTPDA PROC MODULE='FTPD',PARMS='',
//          P1='ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIPA")',
//          FTPDATA=FTPDATA,TCPDATA=TCPDATA
//FTPDA1 EXEC PGM=&MODULE,REGION=OK,TIME=NOLIMIT,
//          PARM='POSIX(ON) ALL31(ON) &P1/&PARMS'
//CEEDUMP DD SYSOUT=*
//SYSFTPD DD DISP=SHR,DSN=TCPIPA.&SYSNAME..TCPparms(&FTPDATA.)
//SYSTCPD DD DISP=SHR,DSN=TCPIPA.&SYSNAME..TCPparms(&TCPDATA.)
//SYSFTSX DD DISP=SHR,DSN=TCPIPMVS.STANDARD.TCPXLBIN
```

ICLI server started task

This procedure is one of four used by both LPARs.

```
//REDICLI6 EXEC PGM=BPXBATCH,TIME=NOLIMIT,REGION=OM,
//          PARM='PGM /usr/sbin/fome46ds -PLAN FOME46D -LOGDIR /usr/sap/
//          RED/icli/icli6 -PORT 5006'
//STDENV DD PATH='/usr/sap/RED/icli/iclienv'
//STEPLIB DD DISP=SHR,DSN=DB7X7.SDSNLOAD
//STDERR DD PATH='/usr/sap/RED/icli/icli6/redicli6.&SYSNAME..err',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRWXU)
//STDOUT DD PATH='/usr/sap/RED/icli/icli6/redicli6.&SYSNAME..out',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRWXU)
//SYSUDUMP DD SYSOUT=*
//SYSMDUMP DD SYSOUT=*
```

A.1.2 z/OS LPAR SC42

In this section, we describe the network settings for LPAR SC42.

File /etc/resolv.conf - SC42

```
TCPIPJobname TCPIPA      ;
Datasetprefix TCPIPA     ;
Messagecase mixed        ;
HostName wtsc42a         ;
DomainOrigin itso.ibm.com ;
NSinterAddr 9.12.2.7     ;
NSportAddr 53            ;
ResolveVia UDP           ;
ResolverTimeout 10       ;
ResolverUdpRetries 1     ;
```

TCPIP profile - SC42

```
; -----
;
;
; Flush the ARP tables every 20 minutes.
;

ARPAGE 20
; GLOBALCONFIG: Provides settings for the entire TCP/IP stack
;
GLOBALCONFIG NOTCPIPSTATISTICS
;
; IPCONFIG: Provides settings for the IP layer of TCP/IP.
;
; Example IPCONFIG for single stack/single system:
;
IPCONFIG
;

    ARPTO 1200          ; In seconds
    DATAGRamfwd
    SOURCEVIPA
    VARSUBNETTING      ; For RIPV2
    PATHMTUDISCOVERY
    SYSPLEXRouting
    DYNAMICXCF 192.168.40.4 255.255.255.0 2
    IGNORERedirect
    REASSEMBlytimeout 15 ; In seconds
    STOPONclawerror
    TTL 60              ; In seconds, but actually Hop count
    SACONFIG COMMUNITY MVSsub1
    ENABLED
    AGENT 161
```

```

VIPADYNAMIC
VIPARANGE DEFINE MOVEABLE DISRUPTIVE 255.255.255.0 172.20.10.0
ENDVIPADYNAMIC
;SOMAXCONN: Specifies maximum length for the connection request queue
;  created by the socket call listen().
;
;SOMAXCONN 10
;
;
;
; TCPCONFIG: Provides settings for the TCP layer of TCP/IP.
;
TCPCONFIG TCPSENDBFRSIZE 16K TCPRCVBUFRSIZE 16K SENDGARBAGE FALSE
;
; UDPCONFIG: Provides settings for the UDP layer of TCP/IP
;
UDPCONFIG UNRESTRICTLOWPORTS
; -----
;
; Reserve low ports for servers
;
TCPCONFIG      RESTRICTLOWPORTS
UDPCONFIG      RESTRICTLOWPORTS
;
; -----
;
; AUTOLOG the following servers
AUTOLOG 5
FTPD JOBNAME FTPDA ; FTP Server
PMAPA ; Portmap Server
MVSNFSSA ;;;;;;; Only for primary
ENDAUTOLOG

;
;
;
; -----
; Reserve ports for the following servers.
;
; NOTES:
;
; A port that is not reserved in this list can be used by any user.
; If you have TCP/IP hosts in your network that reserve ports
; in the range 1-1023 for privileged applications, you should
; reserve them here to prevent users from using them.
;
; The port values below are from RFC 1060, "Assigned Numbers."
;
PORT

```



```

20 TCP OMVS                ; OE FTP Server
    DELAYACKS              ; Delay transmission acknowledgements
21 TCP OMVS                ; OE FTPD control port
23 TCP OMVS                ; OE Telnet Server
80 TCP OMVS                ; OE Web Server
111 TCP OMVS               ; Portmap Server
111 UDP OMVS               ; Portmap Server
135 UDP LLBD               ; NCS Location Broker
161 UDP SNMPD              ; SNMP Agent
162 UDP SNMPQE             ; SNMP Query Engine
512 TCP RXPROCA            ; Remote Execution Server
514 TCP RXPROCA            ; Remote Execution Server
520 UDP ROUTED             ; Routed Server
580 UDP NCPROUT            ; NCPROUTE Server
750 TCP MVSKERB            ; Kerberos
750 UDP MVSKERB            ; Kerberos
751 TCP ADM@SRV            ; Kerberos Admin Server
751 UDP ADM@SRV            ; Kerberos Admin Server
2000 TCP IOASRV            ; OSA/SF Server
2049 UDP MVSNFSSA          ; Our NFS Server

PORTRANGE 10000 2000 TCP OMVS ; TCP 10000 - 11999
PORTRANGE 10000 2000 UDP OMVS ; UDP 10000 - 11999
; -----
;
; Hardware definitions:

DEVICE OSA2880 MPCIPA      PRIROUTER
LINK   OSA2880LNK IPAQENET OSA2880

DEVICE STAVIPA1 VIRTUAL 0
LINK   STAVIPA1L VIRTUAL 0 STAVIPA1

DEVICE IUTIQDEE MPCIPA
LINK   HIPERLEE IPAQIDIO   IUTIQDEE
; -----
;
; HOME internet (IP) addresses of each link in the host.
;
; NOTE:
;
; The IP addresses for the links of an Offload box are specified in
; the LINK statements themselves, and should not be in the HOME list.
;
HOME
172.21.11.1    STAVIPA1L
192.168.60.4   HIPERLEE
192.168.50.4   OSA2880LNK
; -----

```

```

;
; IP routing information for the host. All static IP routes should
; be added here.
;
GATEWAY
    192.168.50 =      OSA2880LNK    1500      0
    192.168.60 =      HIPERLEE     32768      0
;
DEFAULTNET 192.168.50.75 OSA2880LNK 1500      0
; -----
; Turn off all tracing. If tracing is to be used, change the following
; line. To trace the configuration component, for example, change
; the line to ITRACE ON CONFIG 1

ITRACE OFF
;
; -----
; The ASSORTEDPARMS NOFWD will prevent the forwarding of IP packets
; between different networks. If NOFWD is not specified, IP packets
; will be forwarded between networks when this host is a gateway.
;
; Even though RESTRICTLOWPORTS was specified on TCPCONFIG and UDPCONFIG,
; ASSORTEDPARMS default would have been to reset RESTRICTLOWPORTS to off
; So it is respecified here.
; If the TCPCONFIG and UDPCONFIG followed ASSORTEDPARMS, RESTRICTLOWPORT
; would not have to be done twice.

ASSORTEDPARMS
; NOFWD
; RESTRICTLOWPORTS
ENDASSORTEDPARMS
; Start all the defined devices.
;
START OSA2880
START IUTIQDEE

```

OMPROUTE started task - SC42

```

//OMPROUTA PROC
//OMPROUTE EXEC PGM=BPXBATCH,REGION=4096K,TIME=NOLIMIT,
// PARM='PGM /usr/lpp/tcpip/sbin/omproute'
/*      'ENVAR("_CEE_ENVFILE=DD:STDENV")/'
/*
/* PARM=('POSIX(ON)',
/*      'ENVAR("_CEE_ENVFILE=DD:STDENV")/-t1')
/*
/*      Provide environment variables to run with the
/*      desired stack and configuration. As an example,

```

```

/*      the file specified by STDENV could have these
/*      three lines in it:
/*
/*      RESOLVER_CONFIG=// 'SYS1.TCPPARMS(TCPDATA2)'
/*      OMPROUTE_FILE=/u/usernnn/config.tcpcs2
/*      OMPROUTE_DEBUG_FILE=/tmp/logs/omproute.debug
/*
/*      For information on the above environment variables,
/*      refer to the IP CONFIGURATION GUIDE.
/*
//STDENV DD DSN=TCPIPA.&SYSNAME..OMPROUTA.ENVVARS,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*

```

ENVVARS - SC42

```

RESOLVER_CONFIG=// 'TCPIPA.SC42.TCPPARMS(TCPDATA)'
OMPROUTE_FILE=// 'TCPIPA.SC42.TCPPARMS(OMPROUTA)'
OMPROUTE_DEBUG_FILE=/tmp/omprouta.debug

```

OSPF routing parameters - SC42

```

Area      area_number=0.0.0.0
          stub_Area=no
          Authentication_type=none;
ROUTERID=172.21.11.1;
OSPF_Interface IP_Address=172.21.11.1
               Subnet_mask=255.255.255.0
               Router_Priority=1
               Name=STAVIPA1L
               MTU=1500;
OSPF_Interface IP_Address=192.168.60.4
               Subnet_mask=255.255.255.0
               Router_Priority=1
               Name=HIPERLEE
               MTU=16384;
OSPF_Interface IP_Address=192.168.50.4
               Subnet_mask=255.255.255.0
               Router_Priority=0
               Name=OSA2880LNK
               MTU=1500;
Ospf_interface IP_Address=192.168.40.4
               Name=DYNXCF
               Router_Priority=1
               Subnet_mask=255.255.255.0;
OSPF_Interface IP_Address=172.20.10.0
               Subnet_Mask=255.255.255.0
               Router_Priority=1
               Name=VRANGE;
AS_Boundary_routing

```

```

Import_RIP_Routes=YES
Import_Direct_Routes=no
Import_Static_Routes=no;

```

A.1.3 z/OS LPAR SC04

In this section, we describe the network settings for LPAR SC04.

File /etc/resolv.conf - SC04

```

TCPIPJobname TCPIPA      ;
Datasetprefix TCPIPA     ;
Messagecase mixed        ;
HostName wtsc04a         ;
DomainOrigin itso.ibm.com ;
NSinterAddr 9.12.2.7      ;
NSportAddr 53             ;
ResolveVia UDP           ;
ResolverTimeout 10        ;
ResolverUdpRetries 1      ;

```

TCP/IP profile - SC04

```

; -----
;
; Flush the ARP tables every 20 minutes.
;
ARPAGE 20
; GLOBALCONFIG: Provides settings for the entire TCP/IP stack
;
GLOBALCONFIG NOTCPIPSTATISTICS
;
; IPCONFIG: Provides settings for the IP layer of TCP/IP.
;
; Example IPCONFIG for single stack/single system:
;
IPCONFIG
;
; ARPTO 1200           ; In seconds
DATAGRamfwd
SOURCEVIPA
VARSUBNETTING         ; For RIPV2

```

```

PATHMTUDISCOVERY
SYSPLEXRouting
DYNAMICXCF 192.168.40.5 255.255.255.0 2
IGNORERedirect
REASSEMBLYtimeout 15 ; In seconds
STOPONclawerror
TTL 60 ; In seconds, but actually Ho
SACONFIG COMMUNITY MVSSub1
ENABLED
AGENT 161
VIPADYNAMIC
VIPARANGE DEFINE MOVEABLE NONDISRUPTIVE 255.255.255.0 172.20.10.0
ENDVIPADYNAMIC
; SOMAXCONN: Specifies maximum length for the connection request queue
; created by the socket call listen().
;
SOMAXCONN 10
; TCPCONFIG: Provides settings for the TCP layer of TCP/IP.
;
TCPCONFIG TCPSENDBFRSIZE 16K TCPCVBFRSIZE 16K SENDGARBAGE FALSE
;
; UDPCONFIG: Provides settings for the UDP layer of TCP/IP
;
UDPCONFIG UNRESTRICTLOWPORTS
; -----
;
; Reserve low ports for servers
;
TCPCONFIG RESTRICTLOWPORTS
UDPCONFIG RESTRICTLOWPORTS
;
; -----
; AUTOLOG the following servers
AUTOLOG 5
FTPD JOBNAME FTPDA ; FTP Server
PMAPA ; Portmap Server
;MVSNFSSA ;;;; Not on the secondary TCPIP stack;
ENDAUTOLOG
;
;
; -----
; Reserve ports for the following servers.
;
; NOTES:
;
; A port that is not reserved in this list can be used by any user.
; If you have TCP/IP hosts in your network that reserve ports
; in the range 1-1023 for privileged applications, you should
; reserve them here to prevent users from using them.

```

```

;
;   The port values below are from RFC 1060, "Assigned Numbers."
;
PORT
    20 TCP OMVS           ; OE FTP Server
        DELAYACKS        ; Delay transmission acknowledgements
    21 TCP OMVS           ; OE FTPD control port
    23 TCP OMVS           ; OE Telnet Server
    80 TCP OMVS           ; OE Web Server
    111 TCP OMVS          ; Portmap Server
    111 UDP OMVS          ; Portmap Server
    135 UDP LLBD          ; NCS Location Broker
    161 UDP SNMPD         ; SNMP Agent
    162 UDP SNMPQE        ; SNMP Query Engine
    512 TCP RXPROCA       ; Remote Execution Server
    514 TCP RXPROCA       ; Remote Execution Server
    520 UDP ROUTED        ; Routed Server
    580 UDP NCPROUT       ; NCPROUTE Server
    750 TCP MVSKERB       ; Kerberos
    750 UDP MVSKERB       ; Kerberos
    751 TCP ADM@SRV       ; Kerberos Admin Server
    751 UDP ADM@SRV       ; Kerberos Admin Server
    2000 TCP IOASRV       ; OSA/SF Server
    2049 UDP MVSNFSSA     ; NFS Server

PORTRANGE 10000 2000 TCP OMVS ; TCP 10000 - 11999
PORTRANGE 10000 2000 UDP OMVS ; UDP 10000 - 11999
;
; -----
;
; -----
;
; Hardware definitions:

DEVICE OSA2860 MPCIPA PRIROUTER
LINK   OSA2860LNK IPAQNET OSA2860

DEVICE STAVIPA1 VIRTUAL 0
LINK   STAVIPA1L VIRTUAL 0 STAVIPA1
;
; -----
;
; HOME internet (IP) addresses of each link in the host.
;
;
HOME
172.21.10.1 STAVIPA1L
192.168.50.5 OSA2860LNK
;

```

```

;
; IP routing information for the host. All static IP routes should
; be added here.
;
GATEWAY
192.168.50 = OSA2860LNK 1492 0
;
; Default Route - All packets to an unknown destination are routed
; through this route.
;
; Network First Hop Link Name Packet Size Subnet Mask Subnet Value
DEFAULTNET 192.168.50.75 OSA2860LNK 1492 0
ITRACE OFF
;
; -----
; The ASSORTEDPARMS NOFWD will prevent the forwarding of IP packets
; between different networks. If NOFWD is not specified, IP packets
; will be forwarded between networks when this host is a gateway.
;
; Even though RESTRICTLOWPORTS was specified on TCPCONFIG and UDPCONFIG,
; ASSORTEDPARMS default would have been to reset RESTRICTLOWPORTS to off
; So it is respecified here.
; If the TCPCONFIG and UDPCONFIG followed ASSORTEDPARMS, RESTRICTLOWPORT
; would not have to be done twice.

ASSORTEDPARMS
RESTRICTLOWPORTS
ENDASSORTEDPARMS
;
; Start all the defined devices.
;
START OSA2860

```

OMPROUTE started task - SC04

Same as defined for SC42.

ENVVARS - SC04

```

RESOLVER_CONFIG=/'TCPIPA.SC04.TCPPARMS(TCPDATA)'
OMPROUTE_FILE=/'TCPIPA.SC04.TCPPARMS(OMPROUTA)'
OMPROUTE_DEBUG_FILE=/tmp/omprouta.debug

```

OSPF routing parameters - SC04

```

Area      area_number=0.0.0.0
          stub_Area=no
          Authentication_type=none;
ROUTERID=172.21.10.1;
OSPF_Interface IP_Address=172.21.10.1

```

```

Subnet_mask=255.255.255.0
Router_Priority=1
Name=STAVIPA1L
MTU=1500;
OSPF_Interface IP_Address=192.168.50.5
Subnet_mask=255.255.255.0
Router_Priority=0
Name=OSA2860LNK
MTU=1500;
Ospf_Interface IP_Address=192.168.40.5
Name=DYNXCF
Router_Priority=1
Subnet_mask=255.255.255.0
MTU=1500;
OSPF_Interface IP_Address=172.20.10.0
Subnet_Mask=255.255.255.0
Router_Priority=1
Name=VRANGE;
AS_Boundary_routing
Import_RIP_Routes=YES
Import_Direct_Routes=no
Import_Static_Routes=no;

```

A.1.4 OSA-Express Fast Ethernet definitions for z/OS

In this section, we describe the OSA-Express Fast Ethernet definitions we used on z/OS.

SYS1.VTAMLST TRLE definition

```

OSA2880  VBUILD TYPE=TRL
OSA2880P TRLE  LNCTL=MPC,
               READ=2880,
               WRITE=2881,
               DATAPATH=(2882,2883,2884,2885),
               PORTNAME=OSA2880,
               MPCLEVEL=QDIO

```

```

*
*
*
*
*

```


A.2 Linux for zSeries network settings

In this section, we describe the network settings for Linux for zSeries.

A.2.1 Zebra setup - OSPF

```
! *- ospf *-  
!  
! OSPFd sample configuration file  
!  
!  
hostname ospfd  
password zebra  
!enable password please-set-at-here  
!  
!router zebra  
!network 192.168.1.0/24 area 0  
interface hsi1  
  ip ospf cost 5  
  ip ospf priority 5  
interface eth2  
  ip ospf cost 10  
  ip ospf priority 0  
router ospf  
  network 192.168.50.0/24 area 0  
  network 192.168.60.0/24 area 0  
!  
log stdout
```

A.2.2 Zebra setup - Zebra

```
! *- zebra *-  
!  
! zebra sample configuration file  
!  
! $Id: zebra.conf.sample,v 1.14 1999/02/19 17:26:38 developer Exp $  
!  
hostname Router  
password zebra  
enable password zebra  
!  
! Interface's description.  
!  
!interface lo  
interface eth2  
interface hsi1
```

```

!interface hsi1
! description Guest VLAN 2
! bandwidth 10000000
! multicast
! ip ospf cost 1
! description test of desc.
!
!interface sit0
! multicast

!
! Static default route sample.
!
!ip route 0.0.0.0/0 203.181.89.241
!

log file /var/log/zebra.log

```

A.3 Domain Name Server (DNS) definitions

Here are the DNS entries in flat format that we entered for our network.

sapred	A	172.20.10.1
sapnfs	A	172.20.10.2
saprout	A	172.20.10.3
vmlinux6	A	192.168.50.6
wtsc04a	A	172.21.10.1
wtsc42a	A	172.21.11.1
172.20.10.1	PTR	sapred
172.20.10.2	PTR	sapnfs
172.20.10.3	PTR	saprout
192.168.50.6	PTR	vmlinux6
172.210.10.1	PTR	wtsc04a
172.21.11.1	PTR	wtsc42a

File system setup

This appendix includes the NFS server and client procedures with export and attribute files, and file system statements in the BPXPRM member in SYS1.PARMLIB. It also includes the Linux mount commands.

B.1 NFS server procedure

```
//MVSNFSSA PROC MODULE=GFSAMAIN,
//          SYSNFS=SYS1,NFSPRFX=OS390NFS,
//          TCPIP=TCPIPA,
//          TCPDATA=TCPDATA
//*****
//*
//*  NFS SERVER WITH VIPA FAILOVER SUPPORT
//*  VIPA: SAPNFS = 172.20.10.2 ON STACK TCPIPA
//*
//*****
//DEFVIPA  EXEC PGM=EZBXFDVP,REGION=512M,TIME=1440,
//          PARM='POSIX(ON) ALL31(ON) / -p TCPIPA -c 172.20.10.2'
//*
//GFSAMAIN EXEC PGM=&MODULE,REGION=0M,TIME=1440,COND=(4,LT),
//          PARM=(,
//          'ENVAR(" _BPXK_SETIBMOPT_TRANSPORT=TCPIPA")/' )
//SYSTCPD  DD  DISP=SHR,DSN=&TCPIP..&SYSNAME..TCPPARMS(&TCPDATA.)
//STEPLIB  DD  DISP=SHR,DSN=&SYSNFS..NFSLIB
//SYSPRINT DD  SYSOUT=*
//OUTPUT   DD  SYSOUT=*
//SYSERR    DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//NFSATTR   DD  DISP=SHR,DSN=&NFSPRFX..SAPRED.PARMS(ATTRIB)
//EXPORTS   DD  DISP=SHR,DSN=&NFSPRFX..SAPRED.PARMS(EXPORTS)
//NFSLOG1    DD  DISP=SHR,DSN=&NFSPRFX..SAPRED.SERVER.LOG1
//NFSLOG2    DD  DISP=SHR,DSN=&NFSPRFX..SAPRED.SERVER.LOG2
//FHDBASE    DD  DISP=SHR,DSN=&NFSPRFX..SAPRED.FHDBASE1
//FHDBASE2   DD  DISP=SHR,DSN=&NFSPRFX..SAPRED.FHDBASE2
//NFSXLAT    DD  DISP=SHR,DSN=&NFSPRFX..SAPRED.XLAT
```

B.1.1 NFS export file

Following is our export file content:

```
#####
#
#  OS/390 Network File System Server EXPORTS
#
#####
#
/hfs/sapmnt/RED/profile -access=vmlinux6
/hfs/sapmnt/RED/global  -access=vmlinux6
/hfs/sapmnt/RED/AIX/exe  -access=vmlinux6:vmlinux7:vmlinux8:erprisc2
/hfs/sapmnt/RED/Linux/exe -access=vmlinux6
/hfs/sap/trans           -access=vmlinux6
```

B.1.2 NFS attribute file

Following is our attribute file content:

```
space(100,10), blks
norlse
recfm(fb), blksize(0), lrecl(80)
dsorg(ps)
dsntype(pds)
dir(25)
keys(64,0)
recordsize(512,4K)
nonspanned
shareoptions(3,3)
attrtimeout(120), readtimeout(90), writetimeout(30)
text
CRLF
blankstrip
mapleaddot
maplower
retrieve
nofastfilesize
setownerroot
executebitoff
xlat(oemvs311)
nofileextmap
sidefile(OS390NFS.SAPRED.NFS.MAPPING)
security(saf,exports,saf)
pcnfsd
leadswitch
mintimeout(1)
nomaxtimeout
logout(604800) # 60 * 60 * 24 * 7
nfstasks(8,16,8)
restimeout(48,0)
cachewindow(112)
hfs(/hfs)
logicalcache(16M)
bufhigh(32M)
percentsteal(20)
readaheadmax(16K)
maxrdfsleft(32)
smf(none)
sfmax(20)
nochecklist
fn_delimiter(,)
```

B.2 NFS client procedure

```
//NFSCCLNT PROC SYSNFS=SYS1,SYSLE=CEE,NFSPRFX=NFSCCLNT,TCPIP=TCPIPA,
//          TCPDATA=TCPDATA
//*****
//*
//* DFSMS/MVS NETWORK FILE SYSTEM CLIENT START UP PROC(HDZ11SE)
//*
//* CHANGE HISTORY
//* $L41=NFSC,HDZ11SE,960120,SJPLJST: NFS Client support
//*
//*****
//NFSCCLNT EXEC PGM=BPXVCLNY,REGION=OM,TIME=1440,PARM=TP(TCPIPA)
//*
//* &TCPIP..TCPIP.DATA IS TCP/IP V 3.1 OE MVS APPLICATION FEATURE
//* &SYSNFS..NFSLIB IS MVS NETWORK FILE SYSTEM CLIENT TARGET LIBRARY
//* &SYSLE..SCEERUN IS LANGUAGE ENVIRONMENT 1.5 RUNTIME LIBRARY
//*
//SYSTCPD DD DISP=SHR,DSN=&TCPIP..&SYSNAME..TCPPARMS(&TCPDATA.)
//STEPLIB DD DISP=SHR,DSN=&SYSNFS..NFSLIB
//          DD DISP=SHR,DSN=&SYSLE..SCEERUN
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//NFSCMSG1 DD DISP=SHR,DSN=&NFSPRFX..&SYSNAME..LOG1
//NFSCMSG2 DD DISP=SHR,DSN=&NFSPRFX..&SYSNAME..LOG2
```

B.3 Mount commands on Linux /etc/fstab

```
sapnfs:/hfs/sapmnt/RED/global,text,xlat(oemvs311) /sapmnt/RED/global nfs
intr,rsize=8192,wsiz=8192
sapnfs:/hfs/sapmnt/RED/profile,text,xlat(oemvs311) /sapmnt/RED/profile nfs
intr,rsize=8192,wsiz=8192
sapnfs:/hfs/sapmnt/RED/Linux/exe,text,xlat(oemvs311) /sapmnt/RED/Linux/exe nfs
intr,rsize=8192,wsiz=8192
sapnfs:/hfs/sap/trans,text,xlat(oemvs311) /usr/sap/trans nfs
intr,rsize=8192,wsiz=8192
```

B.4 BPXPRM member in SYS1.PARMLIB

Following are the file system statements in our BPXPRM member:

```
FILESYSTYPE TYPE(HFS)           /* Type of file system to start */
                                ENTRYPOINT(GFUAINIT) /* Entry Point of load module */
                                PARM('FSFULL(70,10)') /* IGW023A message issued when
                                                         filesystem is 70% full */
FILESYSTYPE TYPE(ZFS)           /* ZFS */
                                ENTRYPOINT(IOEFSCM)
                                ASNAME(ZFS,'SUB=MSTR')
FILESYSTYPE TYPE(TFS) ENTRYPOINT(BPXTFS)
FILESYSTYPE TYPE(AUTOMNT) ENTRYPOINT(BPXTAMD)
/* files */
VERSION('&SYSR1.')
SYSPLEX(YES)
ROOT FILESYSTEM('WTSCPLX1.SYSPLEX.ROOT')
      TYPE(HFS) /* TYPE OF FILE SYSTEM */
      AUTOMOVE
      MODE(RDWR) /* (OPTIONAL) CAN BE READ OR RDWR.
                  DEFAULT = RDWR */
*/
MOUNT FILESYSTEM('WTSCPLX1.&SYSNAME..SYSTEM.HFS')
      MOUNTPPOINT('/&SYSNAME.')
      NOAUTOMOVE
      TYPE(HFS) MODE(RDWR)
MOUNT FILESYSTEM('OMVS.ZOSR02.&SYSR1..ROOT')
      MOUNTPPOINT('/$VERSION')
      AUTOMOVE
      TYPE(HFS) MODE(RDWR) /* was MODE(READ) */
MOUNT FILESYSTEM('OMVS.&SYSNAME..ETC')
      MOUNTPPOINT('/&SYSNAME./etc')
      NOAUTOMOVE
      TYPE(HFS) MODE(RDWR)
MOUNT FILESYSTEM('OMVS.&SYSNAME..VAR')
      MOUNTPPOINT('/&SYSNAME./var')
      NOAUTOMOVE
      TYPE(HFS) MODE(RDWR)
/* Configure temporary in-storage filesystem for /dev */
MOUNT FILESYSTEM('/&SYSNAME./DEV')
      MOUNTPPOINT('/&SYSNAME./dev')
      NOAUTOMOVE
      TYPE(TFS) MODE(RDWR)
      PARM('-s 10')
/* Configure temporary in-storage filesystem for /tmp */
MOUNT FILESYSTEM('/&SYSNAME./TMP')
      MOUNTPPOINT('/&SYSNAME./tmp')
      NOAUTOMOVE
      TYPE(TFS) MODE(RDWR)
      PARM('-s 500')
```


DB2 and ARM setups

This appendix shows the DB2 ZPARMS and the ARM policy.

C.1 DB2 ZPARMS

C.1.1 ZPARM for subsystem D7X1

```
//D7X1JUZ JOB (999,P0K),'D7X1 INSTALL',CLASS=A,MSGCLASS=T,
// NOTIFY=RC42,TIME=1440,REGION=OM                                00000002
/*JOBPARM SYSAFF=SC42,L=9999                                      00000003
// JCLLIB ORDER=(DB7XU.PROCLIB)                                    00000004
//*****/00010000
//*                                                                */00030000
//*    LICENSED MATERIALS - PROPERTY OF IBM                      */00060000
//*    5675-DB2                                                  */00070000
//*    (C) COPYRIGHT 1982, 2000 IBM CORP.  ALL RIGHTS RESERVED.*/00080000
//*                                                                */00090000
//*    STATUS = VERSION 7                                         */00100000
//*                                                                */00110000
//* FUNCTION = DSNZPARM AND DSNHDECP UPDATES                      */00120000
//*                                                                */00130000
//* PSEUDOCODE =                                                  */00140000
//*    DSNTIZA STEP ASSEMBLE DSN6... MACROS, CREATE DSNZPARM    */00150000
//*    DSNTIZL STEP LINK EDIT DSNZPARM                          */00160000
//*    DSNTLOG STEP UPDATE PASSWORDS                             */00170000
//*    DSNTIZP STEP ASSEMBLE DSNHDECP DATA-ONLY LOAD MODULE    */00180000
//*    DSNTIZQ STEP LINK EDIT DSNHDECP LOAD MODULE              */00190000
//*    DSNTIMQ STEP SMP/E PROCESSING FOR DSNHDECP               */00200000
//*                                                                */00210000
//*    STEP DSNTIMQ MUST BE CUSTOMIZED FOR SMP.  SEE THE NOTES  */00212000
//*    NOTES PRECEDING STEP DSNTIMQ BEFORE RUNNING THIS JOB.    */00214000
//*                                                                */00217000
//*****/00220000
//*                                                                00225000
//DSNTIZA EXEC PGM=ASMA90,PARM='OBJECT,NODECK'                   00230000
//SYSLIB DD DISP=SHR,DSN=DB7X7.SDSNMACS                         00240000
// DD DISP=SHR,DSN=SYS1.MACLIB                                  00260000
//SYSLIN DD DSN=&&LOADSET(DSNTILM1),DISP=(NEW,PASS),              00280000
// UNIT=VIO,                                                     00290000
// SPACE=(800,(50,50,2)),DCB=(BLKSIZE=800)                     00300000
//SYSPRINT DD SYSOUT=*                                           00310000
//SYSUDUMP DD SYSOUT=*                                           00320000
//SYSUT1 DD UNIT=VIO,SPACE=(800,(50,50),,,ROUND)               00330000
//SYSUT2 DD UNIT=VIO,SPACE=(800,(50,50),,,ROUND)               00340000
//SYSUT3 DD UNIT=VIO,SPACE=(800,(50,50),,,ROUND)               00350000
//SYSIN DD *                                                     00360000
DSN6ENV MVS=XA                                                  00370000
DSN6SPRM RESTART,                                              X00370001
ALL,                                                            X00370002
ABEXP=YES,                                                      X00370003
ABIND=YES,                                                      X00370004
```

AUTH=YES,		X00370005
AUTHCACH=1024,		X00370006
BINDNV=BINDADD,		X00370007
BMPTOUT=4,		X00370008
CACHEDYN=YES,		X00370009
CACHEPAC=32768,		X00370010
CACHERAC=32768,		X00370011
CATALOG=DB7XU,		X00370012
CDSSRDEF=1,		X00370013
CHGDC=NO,	<-- WAS YES	X00370014
CONSTOR=YES,	<--- WAS NO	X00370015
DBACRVW=NO,		X00370016
DECDIV3=NO,		X00370017
DEFLTID=IBMUSER,		X00370018
DESCSTAT=NO,		X00370019
DLITOUT=6,		X00370020
DSMAX=6000,	<-- WAS 700	X00370021
EDMBFIT=NO,		X00370022
EDMPOOL=60000,	<-- WAS 13426	X00370023
EDMDSMAX=1048576,		X00370024
EDMDSPAC=100000,	<-- WAS 40960	X00370025
EDPROP=NO,		X00370026
EVALUNC=NO,		X00370027
HOPAUTH=BOTH,		X00370028
IRLMAUT=YES,		X00370029
IRLMPCRC=D7X1IRLM,		X00370030
IRLMSID=I7X1,		X00370031
IRLMRWT=600,	<-- WAS 60	X00370032
IRLMSWT=300,		X00370033
LEMAX=20,		X00370034
MAXRBLK=100000,	<-- WAS 4000	X00370035
MAXKEEPD=12000,	<-- WAS 5000	X00370036
MINDVSCL=NONE,		X00370037
MINSTOR=NO,		X00370038
NPGTHRSH=10,	<-- WAS 0	X00370039
NUMLKTS=1000,		X00370040
NUMLKUS=0,	<-- WAS 10000	X00370041
OPTHINTS=NO,		X00370042
PARAMDEG=0,		X00370043
PARTKEYU=YES,		X00370044
RECALL=YES,		X00370045
RECALLD=120,		X00370046
RELCURHL=YES,		X00370047
RETLWAIT=1,	<-- WAS 0	X00370048
RETVLCFK=NO,		X00370049
RGFCOLID=DSNRGCOL,		X00370050
RGFDBNAM=DSNRGFDB,		X00370051
RGFDEDPL=NO,		X00370052
RGFDEFILT=ACCEPT,		X00370053

	RGFESCP=,		X00370054
	RGFFULLQ=YES,		X00370055
	RGFINSTL=NO,		X00370056
	RGFNMORT=DSN_REGISTER_OBJT,		X00370057
	RGFNMPRT=DSN_REGISTER_APPL,		X00370058
	RRULOCK=NO,		X00370059
	SEQCACH=SEQ,		X00370060
	SEQPRES=NO,		X00370061
	SITETYP=LOCALSITE,		X00370062
	SJTABLES=10,		X00370063
	SRTPOOL=28000,	<-- WAS 1000	X00370064
	STARJOIN=DISABLE,		X00370065
	STATHIST=NONE,		X00370066
	STATROLL=NO,		X00370067
	STATSINT=30,	<-- WAS 1	X00370068
	SUPERRS=YES,		X00370069
	SYSADM=RC42,		X00370070
	SYSADM2=HAIMO,		X00370071
	SYSOPR1=SYSOPR,		X00370072
	SYSOPR2=SYSOPR,		X00370073
	TRKRSITE=NO,		X00370074
	UTIMOUT=3,	<-- WAS 6	X00370075
	XLKUPDLT=YES	<-- WAS NO	00370076
DSN6ARVP	ALCUNIT=BLK,		X00370077
	ARCWRTC=(1,3,4),		X00370078
	ARCWTOR=YES,		X00370079
	ARCPFX1=DB7XU.D7X1.ARCLG1,		X00370080
	ARCPFX2=DB7XU.D7X1.ARCLG2,		X00370081
	ARCRETN=9999,		X00370082
	BLKSIZE=28672,		X00370083
	CATALOG=YES,		X00370084
	COMPACT=NO,		X00370085
	PRIQTY=4320,		X00370086
	PROTECT=NO,		X00370087
	QUIESCE=5,		X00370088
	SECQTY=540,		X00370089
	SVOLARC=NO,		X00370090
	TSTAMP=NO,		X00370091
	UNIT=3390,		X00370092
	UNIT2=		00370093
DSN6LOGP	DEALLCT=(0000),	<--- WAS (0)	X00370094
	MAXARCH=1000,		X00370095
	MAXRTU=2,		X00370096
	OUTBUFF=4000,		X00370097
	OFFLOAD=NO,		X00370097
	TWOACTV=NO,	<-- WAS YES	X00370098
	TWOARCH=NO,	<-- WAS YES	X00370099
	ARC2FRST=NO		00370100
DSN6SYSP	AUDITST=NO,		X00370101

BACKODUR=5,		X00370102
CHKFREQ=50000,		X00370103
CONDBAT=64,		X00370104
CTHREAD=250,	<-- WAS 70	X00370105
DBPROTCL=DRDA,		X00370106
DLDLFREQ=5,		X00370107
DSSTIME=5,		X00370108
EXTRAREQ=100,		X00370109
EXTRASRV=100,		X00370110
EXTSEC=NO,		X00370111
IDBACK=300,	<-- WAS 20	X00370112
IDFORE=40,		X00370113
IDXBPOOL=BPO,		X00370114
LBACKOUT=NO,	<-- WAS AUTO	X00370115
LOBVALA=1000000,	<-- WAS 2048	X00370116
LOBVALS=50000,	<-- WAS 2048	X00370117
LOGAPSTG=10,	<-- WAS 0	X00370118
MAXDBAT=64,		X00370119
MON=NO,		X00370120
MONSIZE=500000,	<--- WAS 8192	X00370121
PCLOSEN=5,		X00370122
PCLOSET=10,		X00370123
RLF=NO,		X00370124
RLFTBL=01,		X00370125
RLFERR=NOLIMIT,		X00370126
RLFAUTH=SYSIBM,		X00370127
ROUTCDE=(1),		X00370128
SMFACCT=(1,2,3),		X00370129
SMFSTAT=(1,3),		X00370130
STATIME=30,		X00370131
STORMXAB=0,		X00370132
STORPROC=D7X1SPAS,		X00370133
STORTIME=180,		X00370134
SYNCVAL=NO,		X00370135
TBSBPOOL=BPO,		X00370136
TRACSTR=NO,		X00370137
TRACTBL=16,		X00370138
URCHKTH=1,	<-- WAS 0	X00370139
URLGWTH=0,		X00370140
WLMENV=		00370141
DDF=AUTO,		X00370142
CMTSTAT=ACTIVE,		X00370143
IDHTOIN=0,		X00370144
RESYNC=2,		X00370145
RLFERRD=NOLIMIT,		X00370146
TCPALVER=NO,		X00370147
MAXTYPE1=0,		X00370148
TCPKPALV=ENABLE,		X00370149
POOLINAC=120		00370150

DSN6FAC

DSN6GRP	DSHARE=YES,	X00370151
	GRPNAM=DB7XU,	X00370152
	MEMBNAM=D7X1,	X00370153
	COORDNTR=NO,	X00370154
	ASSIST=NO,	X00370155
	IMMEDWRI=NO	00370156
END		01370000
//*****		01380000
//*LINK EDIT THE NEW DSNZPARM. THEN PUT LOAD MODULE IN SDSNEXIT.*		01390000
//*****		01400000
//DSNTIZL EXEC PGM=IEWL,PARM='LIST,XREF,LET,RENT',		01410000
// COND=(4,LT)		01420000
//ADSNLOAD DD DISP=SHR,		01430000
// DSN=DB7X7.SDSNLOAD		01440000
// DD DISP=SHR,		01450000
// DSN=DB7X7.ADSNLOAD		01460000
//SYSPUNCH DD DSN=&&LOADSET(DSNTILM1),DISP=(OLD,DELETE)		01470000
//SYSMOD DD DISP=SHR,		01480000
// DSN=DB7X7.SDSNEXIT		01490000
//SYSPRINT DD SYSOUT=*		01500000
//SYSUDUMP DD SYSOUT=*		01510000
//SYSUT1 DD UNIT=VIO,SPACE=(1024,(50,50))		01520000
//SYSLIN DD *		01530000
INCLUDE SYSPUNCH(DSNTILM1)		01540000
INCLUDE ADSNLOAD(DSNZPARM)		01550000
ORDER DSNA		01560000
INCLUDE ADSNLOAD(DSNA)		01570000
INCLUDE ADSNLOAD(DSNFSYSP)		01580000
INCLUDE ADSNLOAD(DSNJARVP)		01590000
INCLUDE ADSNLOAD(DSNJLOGP)		01600000
INCLUDE ADSNLOAD(DSNTSPRM)		01610000
INCLUDE ADSNLOAD(DSNVDIR1)		01620000
INCLUDE ADSNLOAD(DSNZMSTR)		01630000
INCLUDE ADSNLOAD(DSN3DIR1)		01640000
INCLUDE ADSNLOAD(DSN7GRP)		01650000
ENTRY DSNZMSTR		01660000
NAME DSNZPAX1(R)		01670000
//*		01680000
//* CHANGE LOG INVENTORY:		01690000
//* UPDATE BSDS		01700000
//*		01720000
//DSNTLOG EXEC PGM=DSNJU003,COND=(4,LT)		01730000
//STEPLIB DD DISP=SHR,DSN=DB7X7.SDSNLOAD		01740000
//SYSUT1 DD DISP=OLD,DSN=DB7XU.D7X1.BSDS01		01750000
//SYSUT2 DD DISP=OLD,DSN=DB7XU.D7X1.BSDS02		01760000
//SYSPRINT DD SYSOUT=*		01770000
//SYSUDUMP DD SYSOUT=*		01780000
//SYSIN DD *		01790000
DDF LOCATION=DB7X,LUNAME=SCPD7X1,		01790001

```

NOPASSWD,RESPORT=33737,PORT=33736                                01790002
DATASHR ENABLE                                                    01790003
*                                                                    01790004
* !!! DO NOT CHANGE ANY PARAMETERS IN THE GROUP STATEMENT BELOW! 01790005
GROUP GROUPNAM=DB7XU,GROUPMEM=D7X1,MEMBERID=1                    01790006
//*                                                                  02630000
//*****02640000
//* ASSEMBLE AND LINK EDIT DATA-ONLY LOAD MODULE DSNHDECP.      02650000
//* THE FOLLOWING STEPS ARE NEEDED ONLY IF THE                     02660000
//* VALUES ARE CHANGED FROM THOSE WHICH ARE SHIPPED.            02670000
//*****02680000
//DSNTIZP EXEC PGM=ASMA90,PARM='OBJECT,NODECK',COND=(4,LT)       02690000
//SYSLIB DD DISP=SHR,                                             02700000
// DSN=DB7X7.SDSNMACS                                             02710000
//SYSLIN DD DSN=&&LOADSET(DSNHDECA),DISP=(NEW,PASS),UNIT=VIO,    02720000
// SPACE=(80,(50,50,2)),DCB=(BLKSIZE=80)                        02730000
//SYSPRINT DD SYSOUT=*                                           02740000
//SYSUDUMP DD SYSOUT=*                                           02750000
//SYSUT1 DD UNIT=VIO,SPACE=(800,(50,50),,,ROUND)                02760000
//SYSUT2 DD UNIT=VIO,SPACE=(800,(50,50),,,ROUND)                02770000
//SYSUT3 DD UNIT=VIO,SPACE=(800,(50,50),,,ROUND)                02780000
//SYSIN DD *                                                      02790000
DSNHDECM CHARSET=ALPHANUM, X02790001
ASCCSID=819, X02790002
AMCCSID=65534, X02790003
AGCCSID=65534, X02790004
SCCSID=37, X02790005
MCCSID=65534, X02790006
GCCSID=65534, X02790007
USCCSID=367, X02790008
UMCCSID=1208, X02790009
UGCCSID=1200, X02790010
ENSCHHEME=EBCDIC, X02790011
APPENSCH=EBCDIC, X02790012
DATE=ISO, X02790013
DATELEN=0, X02790014
DECARTH=DEC15, X02790015
DECIMAL=PERIOD, X02790016
DEFLANG=IBMCOB, X02790017
DELIM=DEFAULT, X02790018
MIXED=NO, X02790019
SQLDELI=DEFAULT, X02790020
DSQLDELI=APOST, X02790021
SSID=D7XG, X02790022
STDSQL=NO, X02790023
TIME=ISO, X02790024
TIMELEN=0, X02790025
DYNRULS=YES, X02790026
LC_CTYPE=, X02790027

```

COMPAT=OFF	02790028
END	02970000
/*	02980000
/******	02990000
/* LINK EDIT DSNHDECP.	*03000000
/* DSNHDECP IS A DATA-ONLY LOAD MODULE CONTAINING DEFAULT	*03010000
/* VALUES REQUIRED BY DB2 AND APPLICATION PROGRAMS.	* 03020000
/* THIS STEP IS CREATED ONLY WHEN THE DEFAULTS SUPPLIED IN	* 03030000
/* DSNHDECP ARE NOT SUITABLE.	* 03040000
/******	03050000
//DSNTIZQ EXEC PGM=IEWL,PARM='LIST,XREF,LET,RENT',	03060000
// COND=(4,LT)	03070000
//ADSNLOAD DD DISP=SHR,	03080000
// DSN=DB7X7.SDSNEXIT	03090000
// DD DISP=SHR,	03100000
// DSN=DB7X7.ADSNLOAD	03110000
//SYSPUNCH DD DSN=&&LOADSET(DSNHDECA),DISP=(OLD,DELETE)	03120000
//SYSLMOD DD DISP=SHR,	03130000
// DSN=DB7X7.SDSNEXIT	03140000
//SYSPRINT DD SYSOUT=*	03150000
//SYSUDUMP DD SYSOUT=*	03160000
//SYSUT1 DD UNIT=VIO,SPACE=(1024,(50,50))	03170000
//SYSLIN DD *	03180000
INCLUDE SYSPUNCH(DSNHDECA)	03190000
ORDER DSNA	03200000
INCLUDE ADSNLOAD(DSNA)	03210000
INCLUDE ADSNLOAD(DSNARIB)	03220000
INCLUDE ADSNLOAD(DSNHDECP)	03230000
ENTRY DSNHDECP	03240000
MODE AMODE(24),RMODE(24)	03250000
NAME DSNHDECP(R)	03260000
/*	03270000
/*	

C.1.2 ZPARM for subsystem D7X2

//D7X2JUJ JOB (999,P0K),'D7X2 INSTALL',CLASS=A,MSGCLASS=T,	
// NOTIFY=RC48,TIME=1440,REGION=OM	00000002
/*JOBPARM SYSAFF=SC48,L=9999	00000003
// JCLLIB ORDER=(DB7XU.PROCLIB)	00000004
/******	/00010000
/* JOB NAME = DSNTIJUJ	*/00020000
/*	*/00030000
/* DESCRIPTIVE NAME = INSTALLATION JOB STREAM	*/00040000
/*	*/00050000
/* LICENSED MATERIALS - PROPERTY OF IBM	*/00060000


```

/**      5675-DB2                                */00070000
/**      (C) COPYRIGHT 1982, 2000 IBM CORP. ALL RIGHTS RESERVED. */00080000
/**                                           */00090000
/**      STATUS = VERSION 7                      */00100000
/**                                           */00110000
/** FUNCTION = DSNZPARM AND DSNHDECP UPDATES      */00120000
/**                                           */00130000
/** PSEUDOCODE =                                */00140000
/** DSNNTIZA STEP ASSEMBLE DSN6.... MACROS, CREATE DSNZPARM */00150000
/** DSNNTIZL STEP LINK EDIT DSNZPARM              */00160000
/** DSNNTLOG STEP UPDATE PASSWORDS                */00170000
/** DSNNTIZP STEP ASSEMBLE DSNHDECP DATA-ONLY LOAD MODULE */00180000
/** DSNNTIZQ STEP LINK EDIT DSNHDECP LOAD MODULE   */00190000
/** DSNNTIMQ STEP SMP/E PROCESSING FOR DSNHDECP    */00200000
/**                                           */00210000
/** STEP DSNNTIMQ MUST BE CUSTOMIZED FOR SMP. SEE THE NOTES */00212000
/**      NOTES PRECEDING STEP DSNNTIMQ BEFORE RUNNING THIS JOB.*/00214000
/**                                           */00217000
/*******                                          */00220000
/**                                           00225000
//DSNTIZA EXEC PGM=ASMA90,PARM='OBJECT,NODECK'    00230000
//SYSLIB DD DISP=SHR,                            00240000
//      DSN=DB7X7.SDSNMACS                        00250000
//      DD DISP=SHR,                              00260000
//      DSN=SYS1.MACLIB                          00270000
//SYSLIN DD DSN=&&LOADSET(DSNTILM2),DISP=(NEW,PASS), 00280000
//      UNIT=VIO,                                00290000
//      SPACE=(800,(50,50,2)),DCB=(BLKSIZE=800)  00300000
//SYSPRINT DD SYSOUT=*                            00310000
//SYSUDUMP DD SYSOUT=*                            00320000
//SYSUT1 DD UNIT=VIO,SPACE=(800,(50,50),,,ROUND) 00330000
//SYSUT2 DD UNIT=VIO,SPACE=(800,(50,50),,,ROUND) 00340000
//SYSUT3 DD UNIT=VIO,SPACE=(800,(50,50),,,ROUND) 00350000
//SYSIN DD *                                       00360000
      DSN6ENV MVS=XA                               00370000
      DSN6SPRM RESTART,                           X00370001
      ALL,                                         X00370002
      ABEXP=YES,                                  X00370003
      ABIND=YES,                                  X00370004
      AUTH=YES,                                    X00370005
      AUTHCACH=1024,                              X00370006
      BINDNV=BINDADD,                             X00370007
      BMPTOUT=4,                                   X00370008
      CACHEDYN=YES,                               X00370009
      CACHEPAC=32768,                             X00370010
      CACHERAC=32768,                             X00370011
      CATALOG=DB7XU,                              X00370012
      CDSSRDEF=1,                                  X00370013
      CHGDC=NO,                                     X00370014
      <-- WAS YES

```

CONTSTOR=YES,	<-- WAS NO	X00370015
DBACRVW=NO,		X00370016
DECDIV3=NO,		X00370017
DEFLTID=IBMUSER,		X00370018
DESCSTAT=NO,		X00370019
DLITOUT=6,		X00370020
DSMAX=6000,	<-- WAS 700	X00370021
EDMBFIT=NO,		X00370022
EDMPPOOL=60000,	<-- WAS 13426	X00370023
EDMSMAX=1048576,		X00370024
EDMDSPEC=100000,	<-- WAS 40960	X00370025
EDPROP=NO,		X00370026
EVALUNC=NO,		X00370027
HOPAUTH=BOTH,		X00370028
IRLMAUT=YES,		X00370029
IRLMPCRC=D7X2IRLM,		X00370030
IRLMSID=I7X2,		X00370031
IRLMRWT=600,	<--WAS 60	X00370032
IRLMSWT=300,		X00370033
LEMAX=20,		X00370034
MAXRBLK=100000,	<-- WAS 4000	X00370035
MAXKEEPD=12000,	<-- WAS 5000	X00370036
MINDVSCL=NONE,		X00370037
MINSTOR=NO,		X00370038
NPGTHRSH=10,	<--WAS 0	X00370039
NUMLKTS=1000,		X00370040
NUMLKUS=0,	<--- WAS 10000	X00370041
OPTHINTS=NO,		X00370042
PARAMDEG=0,		X00370043
PARTKEYU=YES,		X00370044
RECALL=YES,		X00370045
RECALLD=120,		X00370046
RELCURHL=YES,		X00370047
RETLWAIT=1,	<-- WAS 0	X00370048
RETVLCFK=NO,		X00370049
RGFCOLID=DSNRGCOL,		X00370050
RGFDBNAM=DSNRGFDB,		X00370051
RGFDEDPL=NO,		X00370052
RGFDEFILT=ACCEPT,		X00370053
RGFESCP=,		X00370054
RGFFULLQ=YES,		X00370055
RGFINSTL=NO,		X00370056
RGFNMORT=DSN_REGISTER_OBJT,		X00370057
RGFNMPT=DSN_REGISTER_APPL,		X00370058
RRULOCK=NO,		X00370059
SEQCACH=SEQ,		X00370060
SEQPRES=NO,		X00370061
SITETYP=LOCALSITE,		X00370062
SJTABLES=10,		X00370063

	SRTPOOL=28000,	<-- WAS 1000	X00370064
	STARJOIN=DISABLE,		X00370065
	STATHIST=NONE,		X00370066
	STATROLL=NO,		X00370067
	STATSINT=30,	<-- WAS 1	X00370068
	SUPERRS=YES,		X00370069
	SYSADM=RC48,		X00370070
	SYSADM2=HAIMO,		X00370071
	SYSOPR1=SYSOPR,		X00370072
	SYSOPR2=SYSOPR,		X00370073
	TRKRSITE=NO,		X00370074
	UTIMOUT=3,	<-- WAS 6	X00370075
	XLKUPDLT=YES	<-- WAS NO	00370076
DSN6ARVP	ALCUNIT=BLK,		X00370077
	ARCWRTC=(1,3,4),		X00370078
	ARCWTOR=YES,		X00370079
	ARCPFX1=DB7XU.D7X2.ARCLG1,		X00370080
	ARCPFX2=DB7XU.D7X2.ARCLG2,		X00370081
	ARCRETN=9999,		X00370082
	BLKSIZE=28672,		X00370083
	CATALOG=YES,		X00370084
	COMPACT=NO,		X00370085
	PRIQTY=4320,		X00370086
	PROTECT=NO,		X00370087
	QUIESCE=5,		X00370088
	SECQTY=540,		X00370089
	SVOLARC=NO,		X00370090
	TSTAMP=NO,		X00370091
	UNIT=3390,		X00370092
	UNIT2=		00370093
DSN6LOGP	DEALLCT=(0000),	<-- WAS (0)	X00370094
	MAXARCH=1000,		X00370095
	MAXRTU=2,		X00370096
	OFFLOAD=NO,		X00370097
	OUTBUFF=4000,		X00370097
	TWOACTV=NO,	<-- WAS YES	X00370098
	TWOARCH=NO,	<-- WAS YES	X00370099
	ARC2FRST=NO		00370100
DSN6SYSP	AUDITST=NO,		X00370101
	BACKODUR=5,		X00370102
	CHKFREQ=50000,		X00370103
	CONDBAT=64,		X00370104
	CTHREAD=250,	<-- WAS 70	X00370105
	DBPROTCL=DRDA,		X00370106
	DLDFREQ=5,		X00370107
	DSSTIME=5,		X00370108
	EXTRAREQ=100,		X00370109
	EXTRASRV=100,		X00370110
	EXTSEC=NO,		X00370111

IDBACK=300,	<-- WAS 300	X00370112
IDFORE=40,		X00370113
IDXBPOOL=BP0,		X00370114
LBACKOUT=AUTO,		X00370115
LOBVALA=1000000,	<-- WAS 2048	X00370116
LOBVALS=50000,	<-- WAS 2048	X00370117
LOGAPSTG=10,	<-- WAS 0	X00370118
MAXDBAT=64,		X00370119
MON=NO,		X00370120
MONSIZE=500000,	<- WAS 8192	X00370121
PCLOSEN=5,		X00370122
PCLOSET=10,		X00370123
RLF=NO,		X00370124
RLFTBL=01,		X00370125
RLFERR=NOLIMIT,		X00370126
RLFAUTH=SYSIBM,		X00370127
ROUTCDE=(1),		X00370128
SMFACCT=(1,2,3),	<-- WAS (1)	X00370129
SMFSTAT=(1,3),	<-- WAS YES	X00370130
STATIME=30,		X00370131
STORMXAB=0,		X00370132
STORPROC=D7X2SPAS,		X00370133
STORTIME=180,		X00370134
SYNCVAL=NO,		X00370135
TBSBPOOL=BP0,		X00370136
TRACSTR=NO,		X00370137
TRACTBL=16,		X00370138
URCHKTH=1,	<-- WAS 0	X00370139
URLGWTH=0,		X00370140
WLMENV=		00370141
DSN6FAC DDF=AUTO,		X00370142
CMTSTAT=ACTIVE,		X00370143
IDTHTOIN=0,		X00370144
RESYNC=2,		X00370145
RLFERRD=NOLIMIT,		X00370146
TCPALVER=NO,		X00370147
MAXTYPE1=0,		X00370148
TCPKPALV=ENABLE,		X00370149
POOLINAC=120		00370150
DSN6GRP DSHARE=YES,		X00370151
GRPNAME=DB7XU,		X00370152
MEMBNAME=D7X2,		X00370153
COORDNTR=NO,		X00370154
ASSIST=NO,		X00370155
IMMEDWRI=NO		00370156
END		01370000
/******		01380000
/* LINK EDIT THE NEW DSNZPARM MEMBER. PUT MODULE IN SDSNEXIT. *		01390000
/******		01400000

```

//DSNTIZL EXEC PGM=IEWL,PARM='LIST,XREF,LET,RENT',          01410000
//              COND=(4,LT)                                01420000
//ADSNLOAD DD   DISP=SHR,                                  01430000
//              DSN=DB7X7.SDSNLOAD                          01440000
//              DD   DISP=SHR,                              01450000
//              DSN=DB7X7.ADSNLOAD                          01460000
//SYSPUNCH DD   DSN=&&LOADSET(DSNTILM2),DISP=(OLD,DELETE)    01470000
//SYSLMOD  DD   DISP=SHR,                                  01480000
//              DSN=DB7X7.SDSNEXIT                          01490000
//SYSPRINT DD   SYSOUT=*                                    01500000
//SYSUDUMP DD   SYSOUT=*                                    01510000
//SYSUT1   DD   UNIT=VIO,SPACE=(1024,(50,50))               01520000
//SYSLIN   DD   *                                           01530000
//          INCLUDE SYSPUNCH(DSNTILM2)                      01540000
//          INCLUDE ADSNLOAD(DSNZPARM)                       01550000
//          ORDER DSNAALOAD                                01560000
//          INCLUDE ADSNLOAD(DSNAA)                          01570000
//          INCLUDE ADSNLOAD(DSNFSYSP)                       01580000
//          INCLUDE ADSNLOAD(DSNJARVP)                       01590000
//          INCLUDE ADSNLOAD(DSNJLOGP)                       01600000
//          INCLUDE ADSNLOAD(DSNTSPRM)                       01610000
//          INCLUDE ADSNLOAD(DSNVDIR1)                       01620000
//          INCLUDE ADSNLOAD(DSNZMSTR)                       01630000
//          INCLUDE ADSNLOAD(DSN3DIR1)                       01640000
//          INCLUDE ADSNLOAD(DSN7GRP)                        01650000
//          ENTRY   DSNZMSTR                                  01660000
//          NAME    DSNZPAX2(R)                              01670000
//          *                                                 01680000
//          * CHANGE LOG INVENTORY:                          01690000
//          * UPDATE BSDS                                    01700000
//          *                                                 01720000
//DSNTLOG EXEC PGM=DSNJU003,COND=(4,LT)                      01730000
//STEPLIB  DD   DISP=SHR,DSN=DB7X7.SDSNLOAD                 01740000
//SYSUT1   DD   DISP=OLD,DSN=DB7XU.D7X2.BSDS01              01750000
//SYSUT2   DD   DISP=OLD,DSN=DB7XU.D7X2.BSDS02              01760000
//SYSPRINT DD   SYSOUT=*                                    01770000
//SYSUDUMP DD   SYSOUT=*                                    01780000
//SYSIN    DD   *                                           01790000
//          DDF      LOCATION=DB7X,LUNAME=SCPD7X2,           01790001
//                  NOPASSWD,RESPORT=33738,PORT=33736        01790002
//          DATASHR ENABLE                                    01790003
//          *                                                 01790004
//          * !!! DO NOT CHANGE ANY PARAMETERS IN THE GROUP STATEMENT BELOW! 01790005
//          GROUP GROUPNAM=DB7XU,GROUPMEM=D7X2               01790006
//          *                                                 03270000
//          * *****                                          03280000
//          * DO SMP/E PROCESSING TO TRACK DSNHDECP CHANGES. * 03290000
//          * STEP IS ONLY USED WHEN THE DEFAULT DSNHDECP IS NOT SUITABLE.* 03300000
//          *                                                 03301000

```

```

/* NOTE: THIS STEP MUST BE CUSTOMIZED AS FOLLOWS FOR SMP:      * 03302000
/* 1. LOCATE AND CHANGE THE FOLLOWING STRINGS TO THE VALUES YOU 03303000
/*   SPECIFIED FOR THEM IN JOB DSNTIJAE:                        * 03304000
/*   A.'?SMPPRE?' TO THE PREFIX OF YOUR SMP LIBRARY NAME.      * 03305000
/*   B.'?SMPMLQ?' TO THE MID-LEVEL QUALIFIER OF YOUR SMP CSI* 03306000
/* 2. UPDATE SYSOUT CLASSES AS DESIRED (DEFAULT IS '*')        * 03308000
/******03310000
//DSNTIMQ EXEC PGM=GIMSMP,PARM='CSI=?SMPPRE?.?SMPMLQ?.CSI',    03320000
//          REGION=4096K,COND=(2,LT)                            03325000
//SYSPRINT DD SYSOUT=*                                          03330000
//SYSUDUMP DD SYSOUT=*                                          03335000
//SMPCTL DD *                                                  03340000
//          SET BDY(DSNTARG).                                    03350000
//          JCLIN.                                              03360000
//SMPJCLIN DD DISP=SHR,                                         03370000
//          DSN=DB7XU.D7X2.NEW.SDSNSAMP(DSNTIJUZ)              03380000
/*                                                              03390000

```

C.2 The ARM policy

C.2.1 The ARM policy JCL

```
//ARMPOL JOB (999,P0K),'SAPRES6',CLASS=A,MSGCLASS=T,  
//      NOTIFY=&SYSUID,REGION=4M  
/*JOBPARM SYSAFF=SC42  
/*-----*//  
//S1      EXEC PGM=IXCMIAPU  
//SYSPRINT DD SYSOUT=*  
//SYSIN   DD *  
  
DATA TYPE(ARM)  
  
DEFINE POLICY NAME(ARM01) REPLACE(YES)  
  
RESTART_GROUP(DB7XGRP)  
  TARGET_SYSTEM(SC42,SC04)  
  ELEMENT(DB7XUD7X1)  
    RESTART_ATTEMPTS(3,120)  
    RESTART_TIMEOUT(60)  
    READY_TIMEOUT(900)  
    TERMTYPE(ALLTERM)  
    RESTART_METHOD(ELEMTERM,PERSIST)  
    RESTART_METHOD(SYSTEM,STC,'-D7X1 STA DB2,LIGHT(YES)')  
  ELEMENT(DB7XUD7X2)  
    RESTART_ATTEMPTS(3,120)  
    RESTART_TIMEOUT(60)  
    READY_TIMEOUT(900)  
    TERMTYPE(ALLTERM)  
    RESTART_METHOD(ELEMTERM,PERSIST)  
    RESTART_METHOD(SYSTEM,STC,'-D7X2 STA DB2,LIGHT(YES)')  
  
/*
```


NetView and System Automation for OS/390 basic setup

This appendix contains the following:

- ▶ Definitions for the AOFSAP SDF screen
- ▶ The sample REXX exec SANCHK

D.1 Status Display Facility definition

This section contains the sample SDF panel AOFSAP, the modified SDF tree definition member AOFTSC04, and the modified SDF start screen AOFPSYST.

At a minimum, you might want to use the AOFSAP screen as a base for your own screen developments.

D.1.1 AOFPSYST

```

/****START OF COPYRIGHT NOTICE*****/00010000
/*                                     */00020000
/* Proprietary Statement:             */00030000
/*                                     */00040000
/*      5685-151 5655-137             */00050000
/*      Licensed Materials - Property of IBM */00060000
/*      (C) COPYRIGHT IBM CORP. 1990, 2000 All Rights Reserved. */00070000
/*                                     */00080000
/*      US Government Users Restricted Rights - */00090000
/*      Use, duplication or disclosure restricted by */00100000
/*      GSA ADP Schedule Contract with IBM Corp. */00110000
/*                                     */00120000
/*      STATUS= HKYS100                */00130000
/*                                     */00140000
/****END OF COPYRIGHT NOTICE*****/00150000
/*****/00160000
/* Change Code Vrsn Date Who Description */00170000
/* ----- */00180000
/* $LO=FEATURE,SA21,06JUL00,MIK: Rework for V2R1 */00190000
/*                                     */00200000
/* ***** */00210000
/*                                     */00220000
/* Main system monitoring panel */00230000
/*                                     */00240000
/* - Repeat definitions for each system added */00250000
/* - Remember to put each system on a different line */00260000
/*                                     */00270000
/* - Works with definitions from A0FPXXXX and A0FTXXXX */00280000
/*                                     */00290000
P(SYSTEM,24,80)
TF(01,02,10,WHITE,NORMAL)
TT(SYSTEM)
TF(01,23,58,WHITE,NORMAL)
TT(SA OS/390 - SUPPORT SYSTEMS)
/*                                     */00350000
/* First column is system name */00360000
/*                                     */00370000
TF(03,05,10,T,U)
TT(System)
SF(SC04,05,05,10,N,,SC04)
ST(SC04)
```

SF(SC42,07,05,10,N,,SC42)	00420000
ST(SC42)	00430000
/*	*/ 00500000
/* Second column is the worst subsystem	*/ 00510000
/*	*/ 00520000
TF(03,14,24,T,U)	00530000
TT(Subsystems)	00540000
SF(SC04.APPLIC,05,14,24,N,,SC04,Q1)	00550000
SF(SC42.APPLIC,07,14,24,N,,SC42,Q1)	00560000
/*	*/ 00600000
/* Third column is the worst WTOR	*/ 00610000
/*	*/ 00620000
TF(03,27,34,T,U)	00630000
TT(WTORs)	00640000
SF(SC04.WTOR,05,27,34,N,,SC04,1)	00650000
SF(SC42.WTOR,07,27,34,N,,SC42,1)	00660000
/*	*/ 00700000
/* Fourth column is the worst gateway	*/ 00710000
/*	*/ 00720000
TF(03,37,45,T,U)	00730000
TT(Gateways)	00740000
SF(SC04.GATEWAY,05,37,45,N,,SC04,1)	00750000
SF(SC42.GATEWAY,07,37,45,N,,SC42,1)	00760000
/*	*/ 00800000
/* Fifth column is a set of C I D O indicators for each of the	*/ 00810000
/* product automation packages.	*/ 00820000
/*	*/ 00830000
/* - Each system requires 8 entries here...	*/ 00840000
/*	*/ 00850000
TF(03,48,55,T,U)	00860000
TT(Products)	00870000
/*	*/ 00880000
/* Indicators for SC04	*/ 00890000
/*	*/ 00900000
SF(SC04.CICS,05,48,48,N,,SC04C,)	00910000
ST(C)	00920000
SF(SC04.IMS,05,50,50,N,,SC04I,)	00930000
ST(I)	00940000
SF(SC04.DB2,05,52,52,N,,SC04D,)	00950000
ST(D)	00960000
SF(SC04.OPCERR,05,54,54,N,,SC04O,)	00970000
ST(O)	00980000
/*	*/ 00990000
/* Indicators for SC42	*/ 01000000
/*	*/ 01010000
SF(SC42.CICS,07,48,48,N,,SC42C,)	01020000
ST(C)	01030000
SF(SC42.IMS,07,50,50,N,,SC42I,)	01040000
ST(I)	01050000
SF(SC42.DB2,07,52,52,N,,SC42D,)	01060000
ST(D)	01070000
SF(SC42.OPCERR,07,54,54,N,,SC42O,)	01080000
ST(O)	01090000
/*	*/ 01430000

```

/* Sixth column is a set of P V M B T U indicators for each of the */ 01440000
/* product automation packages. */ 01450000
/* */ 01460000
/* - Each system requires 12 entries here... */ 01470000
/* */ 01480000
TF(03,58,68,T,U) 01490000
TT(System) 01500000
/* */ 01510000
/* Indicators for SC04 */ 01520000
/* */ 01530000
SF(SC04.SP00L,05,58,58,N,,SC04,) 01540000
ST(P) 01550000
SF(SC04.MVSCOMP,05,60,60,N,,SC04,) 01560000
ST(V) 01570000
SF(SC04.MESSAGES,05,62,62,N,,SC040,) 01580000
ST(M) 01590000
SF(SC04.BATCH,05,64,64,N,,SC040,) 01600000
ST(B) 01610000
SF(SC04.ONLINE,05,66,66,N,,SC040,) 01620000
ST(T) 01630000
SF(SC04.TSOUSERS,05,68,68,N,,SC040,) 01640000
ST(U) 01650000
/* */ 01660000
/* Indicators for SC42 */ 01670000
/* */ 01680000
SF(SC42.SP00L,07,58,58,N,,SC42,) 01690000
ST(P) 01700000
SF(SC42.MVSCOMP,07,60,60,N,,SC42,) 01710000
ST(V) 01720000
SF(SC42.MESSAGES,07,62,62,N,,SC420,) 01730000
ST(M) 01740000
SF(SC42.BATCH,07,64,64,N,,SC420,) 01750000
ST(B) 01760000
SF(SC42.ONLINE,07,66,66,N,,SC420,) 01770000
ST(T) 01780000
SF(SC42.TSOUSERS,07,68,68,N,,SC420,) 01790000
ST(U) 01800000
/* ----- The following 2 lines are for SAP HA ----- */ 01810000
SF(SC42.SAP,15,24,74,N,,A0FSAP,) 01820004
ST(S A P High Availability) 01830003
/* */ 01840002
/* */ 02260000
/* PFKey Definitions... */ 02270000
/* */ 02280000
TF(24,01,47,T,NORMAL) 02290000
TT(1=HELP 2=DETAIL 3=RETURN 6=ROLL 8=NEXT SCR) 02300000
TF(24,48,79,T,NORMAL) 02310000
TT( 10=LEFT 11=RIGHT 12=TOP) 02320000
EP 02330000

```

D.1.2 AOFSAP

```

/* **START OF COPYRIGHT NOTICE***** */ 00010030
/* */ 00020030
/* Proprietary Statement: */ 00030030
/* */ 00040030
/* 5655-137 */ 00050030
/* Licensed Materials - Property of IBM */ 00060030
/* (C) COPYRIGHT IBM CORP. 1990, 2000 All Rights Reserved. */ 00070030
/* */ 00080030
/* US Government Users Restricted Rights - */ 00090030
/* Use, duplication or disclosure restricted by */ 00100030
/* GSA ADP Schedule Contract with IBM Corp. */ 00110030
/* */ 00120030
/* STATUS= HKYS100 */ 00130030
/* */ 00140030
/* *END OF COPYRIGHT NOTICE***** */ 00150030
/* ***** */ 00160030
/* */ 00230030
P(AOFSAP,24,80,SYSTEM,SYSTEM, , , ) 00240030
TF(01,12,60,Y,R) 00250047
TT( S A P High Availability ) 00260046
/* */ 00261030
TF(03,01,21,P,NORMAL) 00270044
TT(Local Applications) 00280044
/* */ 00290030
TF(03,40,70,P,NORMAL) 00291044
TT(Moving Applications) 00292033
/* */ 00300030
TF(04,01,06,T,NORMAL) 00350048
TT(SC04) 00360030
TF(04,15,20,T,NORMAL) 00370048
TT(SC42) 00380030
/* */ 00390030
TF(04,40,45,T,NORMAL) 00400048
TT(SC04) 00410032
TF(04,54,59,T,NORMAL) 00411048
TT(SC42) 00412032
/* */ 00413032
TF(05,01,30,T,NORMAL) 00420048
TT(-----) 00430033
TF(05,39,69,T,NORMAL) 00431048
TT(-----) 00432033
/* */ 00480030
SF(SC04.REDB2MSTR,06,01,13,N, , ) 00490044
ST( RED_DB2MSTR) 00500035
SF(SC42.REDB2MSTR,06,15,27,N, , ) 00510044
ST( RED_DB2MSTR) 00520035
SF(SC04.REDB2DBM1,07,01,13,N, , ) 00530044
ST( RED_DB2DBM1) 00540036
SF(SC42.REDB2DBM1,07,15,27,N, , ) 00550044
ST( RED_DB2DBM1) 00560036
SF(SC04.REDB2IRLM,08,01,13,N, , ) 00570044
ST( RED_DB2IRLM) 00580036

```

SF(SC42.REDB2IRLM,08,15,27,N, ,)	00590044
ST(RED_DB2IRLM)	00600036
SF(SC04.REDB2DIST,09,01,13,N, ,)	00610044
ST(RED_DB2DIST)	00620036
SF(SC42.REDB2DIST,09,15,27,N, ,)	00630044
ST(RED_DB2DIST)	00640036
SF(SC04.REDB2SPAS,10,01,13,N, ,)	00650044
ST(RED_DB2SPAS)	00660036
SF(SC42.REDB2SPAS,10,15,27,N, ,)	00670044
ST(RED_DB2SPAS)	00680036
SF(SC04.REDRFC,12,01,13,N, ,)	00690044
ST(RED_RFC)	00700037
SF(SC42.REDRFC,12,15,27,N, ,)	00710044
ST(RED_RFC)	00720037
SF(SC04.REDICLI6,13,01,13,N, ,)	00730044
ST(REDICLI6)	00740037
SF(SC42.REDICLI6,13,15,27,N, ,)	00750044
ST(REDICLI6)	00760037
SF(SC04.REDICLI7,14,01,13,N, ,)	00770044
ST(REDICLI7)	00780037
SF(SC42.REDICLI7,14,15,27,N, ,)	00790044
ST(REDICLI7)	00800037
SF(SC04.REDICLI8,15,01,13,N, ,)	00810044
ST(REDICLI8)	00820037
SF(SC42.REDICLI8,15,15,27,N, ,)	00830044
ST(REDICLI8)	00840037
SF(SC04.REDICLI9,16,01,13,N, ,)	00850044
ST(REDICLI9)	00860037
SF(SC42.REDICLI9,16,15,27,N, ,)	00870044
ST(REDICLI9)	00880037
SF(SC42.APPSRV10,18,15,27,N, ,)	00910044
ST(APPSRV10)	00920038
SF(SC04.APPSRV11,18,01,13,N, ,)	00930049
ST(APPSRV11)	00940038
SF(SC04.SAP_OSCOL,19,01,13,N, ,)	01014049
ST(SAP_OSCOL)	01015038
SF(SC42.SAP_OSCOL,19,15,27,N, ,)	01016049
ST(SAP_OSCOL)	01017038
/*	*/ 01020030
/*	*/ 01021030
SF(SC04.MVSNFSSA,06,40,52,N, ,)	01030544
ST(MVSNFSSA)	01030630
SF(SC04.SAP_RTVIPA,08,40,52,N, ,)	01030944
ST(SAP_RTVIPA)	01031030
SF(SC04.SAP_ROUTER,09,40,52,N, ,)	01031144
ST(SAP_ROUTER)	01031244
SF(SC04.REDVIPA,11,40,52,N, ,)	01031344
ST(RED_VIPA)	01031440
SF(SC04.REDES,12,40,52,N, ,)	01031544
ST(RED_ES)	01031640
SF(SC04.REDSMS,13,40,52,N, ,)	01031744
ST(RED_MS)	01031840
SF(SC04.REDGW,14,40,52,N, ,)	01031944
ST(RED_GW)	01032040

SF(SC04.RED_CO,15,40,52,N, ,)	01032144
ST(RED_CO)	01032240
SF(SC04.RED_SE,16,40,52,N, ,)	01032344
ST(RED_SE)	01032440
SF(SC04.RED_ERS,17,40,52,N, ,)	01032544
ST(RED_ERS)	01032630
SF(SC04.APPSRV06,19,40,52,N, ,)	01032744
ST(APPSRV06)	01032830
SF(SC04.APPSRV07,20,40,52,N, ,)	01032944
ST(APPSRV07)	01033030
SF(SC04.APPSRV08,21,40,52,N, ,)	01033144
ST(APPSRV08)	01033230
/*	*/ 01033330
/*	*/ 01035930
SF(SC42.MVSNFSSA,06,54,66,N, ,)	01036044
ST(MVSNFSSA)	01036140
SF(SC42.SAP_RTVIPA,08,54,66,N, ,)	01036444
ST(SAP_RTVIPA)	01036540
SF(SC42.SAP_ROUTER,09,54,66,N, ,)	01036644
ST(SAP_ROUTER)	01036744
SF(SC42.RED_VIPA,11,54,66,N, ,)	01036844
ST(RED_VIPA)	01036940
SF(SC42.RED_ES,12,54,66,N, ,)	01037044
ST(RED_ES)	01037140
SF(SC42.RED_MS,13,54,66,N, ,)	01037244
ST(RED_MS)	01037340
SF(SC42.RED_GW,14,54,66,N, ,)	01037444
ST(RED_GW)	01037540
SF(SC42.RED_CO,15,54,66,N, ,)	01037644
ST(RED_CO)	01037740
SF(SC42.RED_SE,16,54,66,N, ,)	01037844
ST(RED_SE)	01037940
SF(SC42.RED_ERS,17,54,66,N, ,)	01038044
ST(RED_ERS)	01038140
SF(SC42.APPSRV06,19,54,66,N, ,)	01038244
ST(APPSRV06)	01038340
SF(SC42.APPSRV07,20,54,66,N, ,)	01038444
ST(APPSRV07)	01038540
SF(SC42.APPSRV08,21,54,66,N, ,)	01038644
ST(APPSRV08)	01038740
/*	*/ 01038840
TF(24,01,49,T,NORMAL)	01250037
TT(PF1=HELP 2=DETAIL 3=END 6=ROLL 7=UP 8=DN)	01260030
TF(24,51,79,T,NORMAL)	01270030
TT(9=DEL 10=LF 11=RT 12=TOP)	01280030
PFK9('EVJEAB11 &SNODE,&ROOT.&COMPAPPL,&RV,&DATA')	01290030
EP	01300030

D.1.3 AOFTSC04

```

/* **START OF COPYRIGHT NOTICE***** */ 00010000
/* */ 00020000
/* Proprietary Statement: */ 00030000
/* */ 00040000
/* 5655-137 */ 00050000
/* Licensed Materials - Property of IBM */ 00060000
/* (C) COPYRIGHT IBM CORP. 1990, 2000 All Rights Reserved. */ 00070000
/* */ 00080000
/* US Government Users Restricted Rights - */ 00090000
/* Use, duplication or disclosure restricted by */ 00100000
/* GSA ADP Schedule Contract with IBM Corp. */ 00110000
/* */ 00120000
/* STATUS= HKYS100 */ 00130000
/* */ 00140000
/* *END OF COPYRIGHT NOTICE***** */ 00150000
/* ***** */ 00160000
/* Change-Activity: */ 00170000
/* */ 00180000
/* Change Code Vers Date Who Description */ 00190000
/* ----- */ 00200000
/* */ 00210000
/* $LO=FEATURE,SA21,06JUL00,APC(MIK): Sample rework for V2R1 */ 00220000
/* ***** */ 00230000
1 SC04 00430000
2 SYSTEM 00440000
3 APPLIC 00450000
4 SUBSYS 00460000
2 WTOR 00470000
2 SPOOL 00480000
2 GATEWAY 00490000
2 MVSCOMP 00500000
2 APG 00510000
3 GROUPS 00520000
/* */ 00530000
/* ----- */ 00540000
/* */ 00550000
/* The following subtree is required if the extended CICS product */ 00560000
/* automation has been activated for the system. */ 00570000
/* */ 00580000
2 CICS 00590000
3 CICSHLTH 00600000
3 CICSLMT 00610000
3 CICSAUTO 00620000
3 CICSMSG 00630000
3 CICSSTG 00640000
4 CICSSOS 00650000
4 CICSVIOL 00660000
3 CICSTIMR 00670000
3 CICSTRAN 00680000
3 VTAMACB 00690000
/* */ 00700000
/* ----- */ 00710000

```



```

/*                                                    */ 00720000
/* The following subtree is required if the extended IMS product */ 00730000
/* automation has been activated for the system.                */ 00740000
/*                                                    */ 00750000
2 IMS                                                    00760000
3 IMSMSG                                                00770000
3 IMSARCH                                              00780000
3 IMSMSCL                                              00790000
3 IMSOLDS                                              00800000
3 IMSRECN                                              00810000
3 IMSTIMR                                              00820000
3 IMSTRAN                                              00830000
3 IMSSTRCT                                             00840000
/*                                                    */ 00850000
/* ----- */ 00860000
/* ----- */ 00870000
/* The following subtrees are required if the extended OPC product */ 00880000
/* automation has been activated for the system.                */ 00890000
/* ----- */ 00900000
2 OPCERR                                              00910000
2 BATCH                                              00920000
2 TSUSERS                                              00930000
2 SYSTEM                                              00940000
3 MESSAGES                                             00950000
3 IO                                                  00960000
4 TAPES                                              00970000
4 ONLINE                                             00980000
/* ----- */ 00990000
/* ----- */ 01000000
/* ----- */ 01010000
/* The following subtree is required if the extended DB2 product */ 01020000
/* automation has been activated for the system.                */ 01030000
/* ----- */ 01040000
2 DB2                                                 01050000
3 DB2MSG                                              01060000
/* ----- */ 01061001
/* ----- */ 01062001
/* The following subtree is required if the SAP HA is used      */ 01063008
/* All resource names needs to be customized for your environment. */ 01064008
2 SAP                                                 01070001
3 MVSNFSSA                                            01080005
3 SAP_ROUTER                                          01090005
3 SAP_RTVIPA                                          01100005
3 RED_ERS                                             01110005
3 RED_SE                                              01130005
3 RED_VIPA                                           01140005
3 RED_MS                                              01150005
3 RED_CO                                              01160005
3 RED_GW                                              01170005
3 RED_ES                                              01180005
3 APPSRV06                                           01190005
3 APPSRV07                                           01200005
3 APPSRV08                                           01210005
3 RED_DB2MSTR                                         01220007

```

3 RED_DB2DBM1	01230007
3 RED_DB2IRLM	01240007
3 RED_DB2DIST	01250007
3 RED_DB2SPAS	01260007
3 RED_RFC	01270007
3 REDICLI6	01280007
3 REDICLI7	01281007
3 REDICLI8	01282007
3 REDICLI9	01283007
3 APPSRV10	01290007
3 APPSRV11	01291007
3 SAP_ROUTER	01300007
3 SAP_RTVIPA	01310007
3 SAP_OSCOL	01320007

D.2 Sample REXX procedure

D.2.1 SANCHK

This REXX procedure can be used to display and to clear EXCLUDEs and AVOIDs from Move Groups.

```

/* REXX SANCHK ----- */00001102
/*                               */00001202
/* FUNCTION : Display or CLEAR EXCLUDEs or AVOIDs from MOVE "groups */00001302
/*                               */00001402
/*                               */00001502
/*                               +---- DISPLAY -----+ */00001602
/* SYNTAX : sanchk -----+-----+ */00001702
/*                               +---- CLEAR -----+ */00001802
/*                               */00001902
/* ----- */00002502
Trace 0                                00020000
Arg action .                            00030000
/* Action is either CLEAR or DISPLAY */ 00040000
If action = '' Then action = 'DISPLAY' 00050000
/* Issue processing message ... */      00060000
Address NetVasis ,                      00070000
"PIPE LIT /Gathering data step 1 .../" , 00080000
"| CONS ONLY"                          00090000
/* Find all groups via INGLIST */        00100000
"PIPE (STAGESEP | NAME INGLIST)" ,      00110000
"NETV INGLIST */APG,OUTMODE=LINE" ,     00120000
"| DROP FIRST 3 LINES" ,                00130000
"| DROP LAST 1 LINE" ,                  00140000
/* issue command */00150000
/* remove header */00160000
/* remove trailer */00170000

```

```

" | SEPARATE" , /* split into single msgs */00190000
" | LOC 19.8 / /" , /* only sysplex groups */00200000
" | EDIT WORD 1.1 1 \/\ N WORD 2.1 N" , /* create real name */00210000
" | STEM groups." /* set stem */00220000
00230000
/* Issue processing message ... */00240000
Address NetVAsis ,00250000
"PIPE LIT /Gathering data step 2 .../" ,00260000
" | CONS ONLY"00270000
00280000
cnt = 000290000
errcnt = 000300000
000310000
Do i = 1 to groups.000320000
group = groups.i000330000
000340000
/* Get the group details via INGGROUP */000350000
"PIPE (STAGESEP | NAME INGGROUP)" ,000360000
"NETV INGGROUP "||group||",ACTION=MEMBERS,OUTMODE=LINE" ,000370000
" | DROP FIRST 3 LINES" , /* remove header */000380000
" | TAKE FIRST 2 LINES" , /* get data */000390000
" | SEPARATE" , /* split into single msgs */000400000
" | EDIT WORD 3.* 1" , /* get system names */000410000
" | VAR excl avoid" /* set variable */000420000
000430000
If symbol('excl') = 'LIT' Then excl = ''000440000
If symbol('avoid') = 'LIT' Then avoid = ''000450000
000460000
If excl = '' & avoid = '' Then Iterate i000470000
000480000
000490000
errcnt = errcnt + 1000500000
errgroup.errcnt = group000510000
errdata.errcnt = strip(excl avoid)000520000
000530000
cnt = cnt + 1000540000
outline.cnt = '-----'000550000
cnt = cnt + 1000560000
outline.cnt = 'Group = '||group000570000
cnt = cnt + 1000580000
outline.cnt = ' Excluded = '||excl000590000
cnt = cnt + 1000600000
outline.cnt = ' Avoided = '||avoid000610000
000620000
End i000630000
000640000
If cnt = 0 Then Do000650000
If action = 'CLEAR' Then act = 'clear'000660000
Else act = 'display'000670000
cnt = cnt + 1000680000
outline.cnt = 'Nothing to '||act||' ...'000690000
End000700000
Else Do000710000
cnt = cnt + 1000720000

```

outline.cnt = '-----'	00730000
cnt = cnt + 1	00740000
outline.cnt = 'End of Sanity Check'	00750000
End	00760000
	00770000
outline.0 = cnt	00780000
errgroup.0 = errcnt	00790000
errdata.0 = errcnt	00800000
	00810000
Select	00820000
When action = 'DISPLAY' Then Do	00830000
"PIPE (STAGESEP NAME DISPLAY)" ,	00840000
"STEM outline. COLLECT" ,	00850000
" COLOR YELLOW" ,	00860000
" CONS ONLY"	00870000
End	00880000
When action = 'CLEAR' & errcnt = 0 Then Do	00890000
"PIPE (STAGESEP NAME DISPLAY)" ,	00900000
"STEM outline. COLLECT" ,	00910000
" COLOR YELLOW" ,	00920000
" CONS ONLY"	00930000
End	00940000
When action = 'CLEAR' Then Do	00950000
/* Issue processing message ... */	00960000
Address NetVAsis ,	00970000
"PIPE LIT /Processing CLEAR .../" ,	00980000
" COLOR RED" ,	00990000
" CONS ONLY"	01000000
	01010000
Do i = 1 to errgroup.0	01020000
/* Issue processing message ... */	01030000
Address NetVAsis ,	01040000
"PIPE LIT \Processing CLEAR for " errgroup.i \" ,	01050000
" COLOR RED" ,	01060000
" CONS ONLY"	01070000
	01080000
"PIPE (STAGESEP NAME INGGROUP)" ,	01090000
"NETV INGGROUP " errgroup.i ",ACTION=INCLUDE," ,	01100000
"SYSTEMS=(" errdata.i ")," ,	01110000
"OUTMODE=LINE" ,	01120000
" CONS ONLY"	01130000
End i	01140000
	01150000
/* Issue processing message ... */	01160000
Address NetVAsis ,	01170000
"PIPE LIT /Finished CLEAR processing/" ,	01180000
" COLOR RED" ,	01190000
" CONS ONLY"	01200000
	01210000
End	01220000
Otherwise Nop	01230000
End	01240000
	01250000
Exit	01260000

SAP customization for high availability

E.1 Shell scripts used for automation

This part of the appendix lists all scripts we used in our scenario. The scripts are invoked by System Automation.

startappsrv

This script, shown in Example E-1 on page 270, is used to start a local or remote application server instance. It takes the host name and the instance number of the application server as parameters.

The line starting with `rfcping=` has to be edited to reflect the full path of the `rfcping` utility.

The remote execution must be set up to run without password prompt.

Example: E-1 Script startappsrv

```
#!/bin/sh
if [ $# -lt 2 ]; then
    echo "Usage: $0 <Hostname> <SysNr>"
    exit
fi

ashost=$1
sysnr=$2
rfcping=/usr/sap/RED/rfc/rfcping
### The next two lines actually belong on one line
command="cleanipc $sysnr remove; ./stopsap_${ashost}_${sysnr};
./startsap_${ashost}_${sysnr}"

# The following commands are for debugging purpose only
echo "Actual environment of the process ....."
echo "*****"
env
ulimit -a
echo "*****"

# Check first whether AppServer is already running
# On timeout (rc=3) don't restart, indicate startup failure
$rfcping ping_cnt=1 ashost=$ashost sysnr=$sysnr
rc=$?
if [ rc -eq 0 ]; then
    echo "$_BPX_JOBNAME ACTIVE" > /dev/console
    echo ">>> $ashost $sysnr ACTIVE"
    exit 0
fi
if [ rc -gt 2 ]; then
    sleep 60
    echo "$_BPX_JOBNAME STARTUP FAILED (TIMEOUT)" > /dev/console
    echo ">>> $ashost $sysnr STARTUP FAILED (TIMEOUT)"
    exit 0
fi
# Start remote AppServer
if [ `hostname -s` = $ashost ]
then
    eval $command
else
    /bin/rexec $ashost "$command" &
fi
if [ $? -ne 0 ]; then
    echo "$_BPX_JOBNAME STARTUP FAILED" > /dev/console
    echo ">>> $ashost $sysnr STARTUP FAILED"
    exit 0
fi
```

```

# Verify that startup was successful, retry for 120 seconds
retries=0
while [ retries -lt 12 ]
do
    $rfcping ping_cnt=1 ahost=$ashost sysnr=$sysnr
    if [ $? -eq 0 ]; then
        echo "$_BPX_JOBNAME ACTIVE" > /dev/console
        echo ">>> $ashost $sysnr ACTIVE"
        exit 0
    fi
    /bin/sleep 10
    retries=`expr $retries + 1`
done

# Obviously startup has failed
echo "$_BPX_JOBNAME STARTUP FAILED" > /dev/console
echo ">>> $ashost $sysnr STARTUP FAILED"

```

stopappsrv

This script, shown in Example E-2, is used to stop a local or remote application server instance. It takes the host name and the instance number of the application server as parameters.

The remote execution must be set up to run without password prompt.

Example: E-2 Script stopappsrv

```

#!/bin/sh
if [ $# -lt 2 ]; then
    echo "Usage: $0 <Hostname> <SysNr>"
    exit
fi

ashost=$1
sysnr=$2
command="stopsap_${ashost}_${sysnr}"

# Start remote AppServer
if [ `hostname -s` = $ashost ]
then
    eval $command
else
    /bin/rexec $ashost $command
fi

```

checkappsrv

This script, shown in Example E-3, is used to start the monitor for a local or remote application server instance. It takes the host name and the instance number of the application server as parameters.

The line starting with `rfcping=` has to be edited to reflect the full path of the `rfcping` utility. In addition, the `cd` command has to be adapted.

Example: E-3 Script checkappsrv

```
#!/bin/sh
if [ $# -lt 2 ]; then
    echo "Usage: $0 <Hostname> <SysNr>"
    exit
fi

ashost=$1
sysnr=$2
rfcping=/usr/sap/RED/rfc/rfcping

cd /usr/sap/RED/rfc
/bin/ln -sf $rfcping ./rfcping_${ashost}_${sysnr}
### The next two lines actually belong on one line
./rfcping_${ashost}_${sysnr} ping_cnt=-1 ping_interval=10 ashost=$ashost
sysnr=$sysnr > /dev/null
echo "$_BPX_JOBNAME ENDED" > /dev/console
```

startsap_em00

The shell script shown in Example E-4 is used to start the components of the enqueue instance. The component is identified with one of the following abbreviations: MS, ES, ERS, CO, SE, GW. Furthermore, the `CHECK` option performs a health check on the enqueue server.

The lines starting with `DIR_INSTANCE=`, `DIR_EXECUTABLE=`, and `PROFILE=` are to be adapted.

Example: E-4 Script startsap_em00

```
#!/bin/sh
DIR_INSTANCE=/usr/sap/RED/EM00
DIR_EXECUTABLE=/usr/sap/RED/SYS/exe/run
PROFILE=/usr/sap/RED/SYS/profile/RED_EM00

_ES=es.sapRED_EM00
_MS=ms.sapRED_EM00
_CO=co.sapRED_EM00
_SE=se.sapRED_EM00
_GW=gw.sapRED_EM00
_ERS=ers.sapRED_EM00
```

```

cd $DIR_INSTANCE/work

case "$1" in

    MS) rm -f $_MS
        ln -s -f $DIR_EXECUTABLE/msg_server $_MS
        $_MS pf=$PROFILE
        ;;

    ES) rm -f $_ES
        ln -s -f $DIR_EXECUTABLE/enserver $_ES
        $_ES pf=$PROFILE
        ;;

    ERS) rm -f $_ERS
        ln -s -f $DIR_EXECUTABLE/enrepsvr $_ERS
        $_ERS pf=$PROFILE
        ;;

    CO) rm -f $_CO
        ln -s -f $DIR_EXECUTABLE/rslgcoll $_CO
        $_CO -F pf=$PROFILE
        if [ "$?" -gt 0 ]
        then echo "$_BPX_JOBNAME COLLECTOR NOT STARTABLE" > /dev/console
            exit 8
        fi
        ;;

    SE) rm -f $_SE
        ln -s -f $DIR_EXECUTABLE/rslgend $_SE
        $_SE -F pf=$PROFILE
        ;;

    GW) rm -f $_GW
        ln -s -f $DIR_EXECUTABLE/gwrd $_GW
        $_GW pf=$PROFILE
        ;;

    CHECK) $DIR_EXECUTABLE/ensmon pf=$PROFILE 1
        if [ "$?" -gt 0 ]
        then echo "$_BPX_JOBNAME MONITORING FAILED" > /dev/console
            fi
            exit $?
            ;;

    *) echo "Missing or wrong parameter $1"
        echo "Usage: $0 {MS|ES|ERS|CO|SE|GW|CHECK}"
        exit 16

esac
echo "$_BPX_JOBNAME ENDED" > /dev/console

```

start_rfcoscol

The shell script shown in Example E-5 is used to start rfcoscol.

The **cd** command has to be adapted to locate the directory which contains the RFC definition file saprfc.ini.

Example: E-5 Script start_rfcoscol

```
#!/bin/sh
export RFCOSCOL_RETRY=1
export SAP_CODEPAGE=1100
cd /usr/sap/RED/rfc
$DIR_LIBRARY/rfcoscol -DRED_`hostname -s`
echo "$_BPX_JOBNAME ENDED" > /dev/console
```

saprfc.ini

Example E-6 shows the RFC definition file saprfc.ini.

Example: E-6 Definition file saprfc.ini

```
DEST=RED_wtsc42a
TYPE=R
PROGID=wtsc42a.rfcoscol
GWHOST=sapred
GWSERV=sapgw00
RFC_TRACE=0
#
DEST=RED_wtsc04a
TYPE=R
PROGID=wtsc04a.rfcoscol
GWHOST=sapred
GWSERV=sapgw00
RFC_TRACE=0
#
DEST=RED_vmlinux8
TYPE=R
PROGID=vmlinux8.rfcoscol
GWHOST=vmlinux8
GWSERV=sapgw00
RFC_TRACE=0
```

E.2 SAP profiles

This section shows the SAP profiles we used in our configuration. The DEFAULT.PFL and the RED_EM00 are the ones that are relevant for the new

enqueue instance with the standalone enqueue server. The instance profile is simply shown for completeness.

RED_EM00

Example E-7 shows the profile of the enqueue instance.

Example: E-7 Instance profile RED_EM00

```
# Profile for stand-alone enqueue server, message server,  
# gateway, syslog collector, etc.
```

```
SAPSYSTEMNAME = RED  
INSTANCE_NAME = EM00  
SAPSYSTEM = 00
```

```
enqueue/process_location = LOCAL  
enqueue/server/replication = true  
enqueue/server/threadcount = 3  
enqueue/encni/repl_port = 6000  
enqueue/backup_file = $(DIR_GLOBAL)/ENQBCK
```

```
ipc/shm_psize_16 = 0  
ipc/shm_psize_26 = 0  
ipc/shm_psize_66 = 0
```

DEFAULT.PFL

The DEFAULT.PFL profile reflects the changes that are necessary for the enqueue instance; see Example E-8.

Furthermore, the SAPDBHOST variable has been modified as forward reference to an instance profile parameter. This is useful in a data sharing environment.

Setting the WLM parameter in the DEFAULT.PFL enables the registration for all z/OS application server instances and the components of the enqueue instance.

Example: E-8 Profile DEFAULT.PFL

```
#.*****  
#.*  
#.*      Default profile DEFAULT  
#.*  
#.*      Version              = 000005  
#.*      Generated by user = SAPRES5  
#.*      Generated on = 06.13.2002 , 17:02:24  
#.*  
#.*****  
SAPSYSTEMNAME = RED  
SAPDBHOST = $(dbs/db2/hosttcp)
```

```
# Changes for HA enqueue server
rdisp/mshost = sapred
rdisp/sna_gateway = sapred
rdisp/sna_gw_service = sapgw00
rdisp/vbname = $(rdisp/myname)
# rdisp/enqname
rdisp/btcname = $(rdisp/myname)
enqueue/process_location = REMOTESA
enqueue/serverhost = sapred
enqueue/serverinst = 00
# End of changes for HA enqueue server

rslg/collect_daemon/listen_port = 3900
rslg/collect_daemon/talk_port = 4000
rdisp/bufrefmode = sendon,exeauto
# Activate WLM Management
rdisp/prio/wlm/enabled = 1
```

DEFAULT.CentrallInstance

For completeness, we also include here the original DEFAULT.PFL; see Example E-9. We saved it as DEFAULT.CentrallInstance. Restoring it as DEFAULT.PFL would allow us to bring up the central instance instead of the enqueue instance.

Example: E-9 Original profile DEFAULT.CentrallInstance

```
#.*****
#.*
#.*      Default profile DEFAULT
#.*
#.*      Version              = 000002
#.*      Generated by user = DDIC
#.*      Generated on = 04.26.2002 , 15:49:27
#.*
#.******
SAPSYSTEMNAME = RED
SAPDBHOST = wtsc42a
rdisp/mshost = wtsc42a
rdisp/sna_gateway = wtsc42a
rdisp/sna_gw_service = sapgw00
rdisp/vbname = wtsc42a_RED_00
rdisp/enqname = wtsc42a_RED_00
rdisp/btcname = wtsc42a_RED_00
rslg/collect_daemon/listen_port = 3900
rslg/collect_daemon/talk_port = 4000
rdisp/bufrefmode = sendon,exeauto
```

RED_D10_wtsc42a

The profile of the z/OS application server instance (the D10 instance) is shown in Example E-10. (The D11 instance on the second system is similar.)

Example: E-10 Instance profile RED_D10_wtsc42a

```
#.*****
#.*
#.*      Instance profile RED_D10_WTSC42A
#.*
#.*      Version              = 000004
#.*      Generated by user = SAPRES5
#.*      Generated on = 06.13.2002 , 17:02:26
#.*
#.******
```

```
# Changes made for the HA solution
enqueue/con_retries = 120
```

```
# Changes made for z/OS performance tuning
abap/heap_area_total = 2000000000
ztta/roll_area = 6500000
ztta/roll_extension = 2000000000
ztta/roll_first = 1
ES/TABLE = SHM_SEGS
ES/SHM_SEG_SIZE = 0
em/initial_size_MB = 512
em/address_space_MB = 512
em/max_size_MB = 8192
abap/heap_area_dia = 500000000
abap/heap_area_nondia = 500000000
rdisp/PG_MAXFS = 32768
rdisp/PG_SHM = 4096
rdisp/ROLL_MAXFS = 8192
rdisp/ROLL_SHM = 8192
```

```
SAPSYSTEMNAME = RED
INSTANCE_NAME = D10
SAPSYSTEM = 10
rdisp/wp_no_vb = 2
rdisp/wp_no_btc = 2
rdisp/wp_no_vb2 = 1
rdisp/wp_no_dia = 10
zcsa/system_language = E
# DB2 special parameters
dbms/type = db2
rsdb/icli_library = /usr/sap/RED/SYS/exe/run/ibmiclic.dll
rsdb/db2jcl_library = /usr/sap/RED/SYS/exe/run/db2jcllib.dll
rsdb/max_blocking_factor = 10
```

```
rsdb/max_in_blocking_factor = 35
rsdb/min_blocking_factor = 3
rsdb/min_in_blocking_factor = 6
rsdb/prefer_fix_blocking = 1
rsdb/prefer_union_all = 1
rsdb/prefer_in_itab_opt = 1
abap/no_dsqli95 = "DB2"
dbs/db2/ssid = D7XG
dbs/db2/hosttcp = wtsc42a
# rslg/send_daemon/listen_port - Log Sender Listen Port
rslg/send_daemon/listen_port = 3710
# rslg/send_daemon/talk_port - Log Sender Talk Port
rslg/send_daemon/talk_port = 1310
# rsdb/cua/buffersize - Size of CUA Buffer
rsdb/cua/buffersize = 5000
# zcsa/presentation_buffer_area - Size of the buffer allocated for screens
zcsa/presentation_buffer_area = 20000000
# zcsa/table_buffer_area - Size of Resident-Table Buffer
zcsa/table_buffer_area = 50000000
# zcsa/db_max_buftab - Directory Entries in Table Buffer
zcsa/db_max_buftab = 10000
# rtbb/buffer_length - Size of partial table buffers
rtbb/buffer_length = 20000
# rtbb/max_tables - Number of Buffer Directory Entries
rtbb/max_tables = 500

ipc/shm_psize_10 = 88000000
ipc/shm_psize_40 = 70000000
```

Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

<ftp://www.redbooks.ibm.com/redbooks/SG246836>

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 redbook form number, SG246836.

Using the Web material

The additional Web material that accompanies this redbook includes the following files:

<i>File name</i>	<i>Description</i>
Pdb2xmit.bin	Sample SA/390 Policy Database in TSO XMIT format.

Sapmsgux.txt	Sample Automation Table for SAP HA
Sanchk.txt	Sample REXX program to check for and to clear Move Group EXCLUDEs or AVOIDs.
Saprfc.ini	Sample RFC definition file.
Checkappsrv	Sample script used to start the SAP monitor for local and remote application servers.
Default.pfl	Default profile for the new enqueue instance.
Red_EM00	Instance profile for the new enqueue instance.
Start_rfcoscol	Sample shell script used to start rfcoscol.
Startappsrv	Sample shell script used to start a local or remote application server instance.
Startsap_em00	Sample shell script used to start the components of the new enqueue instance.
Stopappsrv	Sample shell script used to stop a local or remote application server instance.
SAP.zip	Zipped file including the above files

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 282.

- ▶ *SAP on DB2 UDB for OS/390 and z/OS - Implementing Application Servers for Linux on zSeries*, SG24-6847
- ▶ *SAP R/3 on DB2 UDB for OS/390: Database Availability Considerations*, SG24-5690

Other resources

These publications are also relevant as further information sources:

- ▶ *Network File System Customization and Operation*, SC26-7417
- ▶ *SAP R/3 on DB2 UDB for OS/390 and z/OS: Planning Guide*, SC33-7966
- ▶ *SAP R/3 on DB2 for OS/390: Connectivity Guide*, SC33-7966
- ▶ *R/3 Installation on OS/390 Unix System Services*, Release 4.6C SR2
- ▶ *DB2 UDB for OS/390 and z/OS Version 7: Data Sharing Planning and Administration*, SC26-9935
- ▶ *System Automation for OS/390: Customizing and Programming*, SC33-7035
- ▶ *System Automation for OS/390: Programmer's Reference*, SC33-7043
- ▶ *SAP R/3 Installation on Linux for zSeries: IBM DB2 UDB for OS/390 and z/OS*, Release 4.6C SR2 (product publication)

SAP notes

The following SAP notes are referenced in this redbook:

- ▶ 509529: DB2/390: Changing the DB2 host proactively
- ▶ 402078: DB2/390: New failover support with 6.10 (kernel)
- ▶ 026317: DB2/390: Set up LOGON group for automatic load balancing

For further information, refer to the SAP Web site:

www.sap.com

Referenced Web sites

These Web sites are also relevant as further information sources:

- ▶ System Automation for OS/390 information
<http://www.ibm.com/servers/eserver/zseries/software/sa>
- ▶ White paper
System Automation for OS/390: *Enhancements for OS/390 UNIX System Services Automation*
<http://www.ibm.com/servers/eserver/zseries/software/sa/pdf/inguss.pdf>
- ▶ Tips for (SA/390) startup and shutdown problems
<http://www.ibm.com/servers/eserver/zseries/software/sa/adds/hint02.html>

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Index

A

- AIX 46
- AOFSAP 261
- application design 24
- application groups 95, 111, 117, 119, 121, 129
- application server 76
- applications 93, 96, 108, 113, 116, 118, 120, 128
- ARM 53–54, 241
- automation 1
- automation table 130

B

- BPXPRM member 239

C

- central instance 12–13, 56
- change management 209
 - rolling kernel upgrade 212
 - rolling upgrade of the ICLI client 213
 - rolling upgrade of the ICLI server 214
 - updating DB2 and z/OS 215
 - updating the enqueue server 211
 - updating the ICLI client and server 213
 - updating the replication server 211
 - updating the SAP dispatcher 211
 - updating the SAP kernel 210

D

- database server 23, 57, 92
 - data sharing 24
 - non-data sharing 24
- DB2
 - ARM policy 241, 255
 - updating 215
 - ZPARMS 241–242
- DB2 setup 52
 - ARM policy 52
 - data sharing 52
- DFS SMB 21, 128
- DISPINFO 194
- DISRUPTIVE 42
- Domain Name Server 234

E

- enqueue replication server 14, 73
- enqueue server 12–14, 42, 73
 - failover 14
 - recovery 14

F

- failure scenarios
 - enqueue server 163
 - ICLI server 174
 - LPAR 185
 - message server 170
 - NFS server 177
 - TCP/IP stack 180
- file system 21, 46–48, 235
 - definitions 51
 - SAP directories 48
 - shared HFS 21
 - sizing considerations 51

G

- gateway 12, 73

H

- HFS 21
 - shared 21, 46
- high availability 1
 - definitions 4
 - degrees 4
 - objectives 6
 - SAP solution 8
- HiperSockets 40, 45

I

- ICLI 37, 53, 72, 209
 - client
 - rolling upgrade 213
 - updating 213
 - server 222
 - rolling upgrade 214
 - updating 213
 - updating with different protocol version 215

INGINFO 195

L

Linux on zSeries 44
local applications 113, 120

M

message server 12, 73
MOVEABLE 42

N

naming conventions 37
NetView 54, 87, 257
 region size 87
network configuration 40
network setup 217
 DNS 234
 Linux for zSeries 233
 OSA-Express Fast Ethernet 232
 z/OS 217
 Zebra 233
NFS

 client 44, 235, 238
 failover 21
 server 20, 43, 46, 116, 235–236

NMC 54

O

OMPROUTE 44
OSA-Express Fast Ethernet 40, 232
OSPF 16, 44, 227, 231
outages 5
 planned 5
 unplanned 5

P

problem determination 191
 application problems 199
 before giving up 198
 DB2 and SAP connections 205
 get help from the Web 199
 if nothing happens 198
 MPF 193
 NetView netlog 192
 network problems 201
 active connection 204
 devices 203

dynamic VIPA 204
 routing tables and OSPF 204
shared HFS and NFS 205
UNIX messages 197
where to look in System Automation 193
z/OS syslog 192

R

Redbooks Web site 282
 Contact us xiv
REXX procedures 266
RFCOSCOL 79
router 40

S

SAMBA 21
SAP high availability 11
 architecture 11
 configuration 34
 hardware 34
 network 40
 software 36
 customization 65
 enqueue instance 66
 SAP 65, 269
 System Automation 71, 85
 failure scenarios 25
 full-blown HA 30
 old style central instance 26
 naming conventions 37
 planning 33
 SAP profiles 274
SAP installation 55
 additional non-z/OS Dialog Instances 60
 additional z/OS Dialog Instances 58
 central instance 56
 database server instance 57
 file systems 62
 logon groups 63
 new SAP system 55
 prepare an existing SAP system for HA 61
 SAP license 62
SAP kernel 209
 rolling kernel upgrade 212
 updating the dispatcher 211
 updating the SAP kernel 210
SAPOSCOL 78
SAPROUTER 43, 81, 117

- SD 130
- SDF 54, 87, 258
- shutdown and restart of an LPAR 153
- SOURCEVIPA 16, 42
- startup of all LPARs one after the other 152
- stop and start of the entire SAP RED system 147
- superior groups 122
- syslog collector 12
- syslog sender 73
- System Automation 54, 85, 257
 - application groups 127
 - applications 126
 - before you start 86
 - classes 90
 - database server 92
 - enqueue instance 96
 - initialization defaults 86
 - MAXFILEPROC 89
 - overall picture 125
 - policy 128
 - problem determination 191
 - application problems 199
 - DB2 and SAP connections 205
 - if nothing happens 198
 - lost 198
 - NetView netlog 192
 - network trouble-shooting 201
 - shared HFS and NFS 205
 - web 199
 - z/OS system log 192
 - SAP-related resources 89
 - SDF 87
 - shell scripts 269
 - syslog 89

T

- TCP/IP 16, 41, 221
- test methodology 137
 - analyze problems 147
 - expected behavior 137
 - purpose 137
 - setup 137
 - verification of resource status 137
- test scenarios 134
 - classification 134
 - planned outage 134
 - controlled operator intervention 134
 - planned shutdown 135

- startup of the entire sysplex 135
- test methodology 137
- unplanned outage 135
 - failure of a network component 136
 - failure of an LPAR 136
 - failure of an SAP component 135

V

- verification procedures 133
- VIPA 12
 - dynamic 16, 20, 42
 - static 16, 20, 42
- VIPARANGE 42, 224

W

- Windows 2000 46

Z

- z/OS
 - updating 215
- Zebra 46



SAP on DB2 UDB for OS/390 and z/OS: High Availability Solution Using System Automation

(0.5" spine)
0.475" <-> 0.875"
250 <-> 459 pages



SAP on DB2 UDB for OS/390 and z/OS: High Availability Solution Using System Automation



New enqueue instance implementation

System Automation customizations for SAP high availability

Failover scenarios

This IBM Redbook explains the SAP high availability with System Automation solution. This solution provides the means for fully automating the management of all SAP components and related products running on OS/390, z/OS, AIX, Windows, or Linux. The automation software monitors all resources and controls the restart and/or takeover of failing components, thereby ensuring near continuous availability of the SAP system.

The book describes the following:

Requirements and objectives of high availability and automation for SAP

Architecture of the SAP high availability solution

Planning and preparation steps

SAP customization for high availability

System Automation customizations for SAP high availability

Failover scenarios

Problem determination

Change management

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