

SAP on z Systems



Business Continuity for SAP on IBM z Systems

Edition 12/2016

SAP on z Systems



Business Continuity for SAP on IBM z Systems

Edition 12/2016

Note

Before using this document and the product it supports, be sure to read the information in "Notices" on page 323.

Edition notice

This edition of *Business Continuity for SAP on IBM z Systems* applies to the following software components and to all subsequent releases and modifications until otherwise indicated in new editions:

- SAP NetWeaver 7.5 and 7.4 based on 7.4x Downward Compatible Kernel (DCK), and SAP NetWeaver 7.1, 7.2, and 7.3 based on 7.2x DCK levels
- z/OS Release 2.1 and higher supported z/OS releases
- AIX Release 6.1 and higher supported versions
- Linux on z Systems
- Linux on System x
- Windows Server 2008 R2 and higher supported versions
- IBM DB2 11 for z/OS
- IBM DB2 12 for z/OS
- IBM Tivoli System Automation for z/OS version 3.5
- IBM Tivoli System Automation for Multiplatforms version 4.1.0 and higher supported releases

IBM welcomes your comments. A form for your comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Deutschland Research and Development GmbH
Department 3282
Schönaicher Straße 220
D-71032 Böblingen
Federal Republic of Germany

Email: s390id@de.ibm.com

World Wide Web:
developerWorks: SAP on IBM z Systems Community
IBM mainframe software for IBM z Systems

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

© Copyright IBM Corporation 2004, 2016.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
--------------------------	------------

Tables	ix
-------------------------	-----------

Summary of changes in edition 12/2016	xi
--	-----------

Summary of changes in edition 12/2015	xi
---	----

Summary of changes in edition 12/2014.	xii
--	-----

About this document	xiii
--------------------------------------	-------------

Content of this document	xiv
------------------------------------	-----

Chapter 1. Introduction of high availability and automation for SAP . . . 1

High availability definitions	3
---	---

Degrees of availability	3
-----------------------------------	---

Types of outages	3
----------------------------	---

The autonomic computing self-healing technologies of Tivoli System Automation	4
---	---

High availability and automation objectives for SAP	5
---	---

Overview of the high availability solution for SAP.	6
---	---

Conventions and terminology used in this document	8
---	---

Chapter 2. SAP availability benefits provided by IBM z Systems 11

Features of z/OS	11
----------------------------	----

Availability features and benefits with z Systems	
---	--

Parallel Sysplex	12
----------------------------	----

DB2 data sharing on z Systems Parallel Sysplex	13
--	----

DB2 connection failover	15
-----------------------------------	----

DB2 connection failover for ABAP instances	16
--	----

DB2 connection failover for Java instances	18
--	----

Building blocks for DB2 connection failover	19
---	----

Chapter 3. Planning for high availability for SAP 21

Possible preparation stages of high availability for SAP	21
--	----

How to decide which System Automation option to implement.	24
--	----

Option 1: Automation using System Automation for z/OS only	24
--	----

Option 2: Automation by System Automation products with central components on AIX or Linux	26
--	----

Technologies on IBM z Systems for highly available SAP solutions	28
--	----

Disaster recovery	32
-----------------------------	----

Data sharing considerations for disaster recovery	32
---	----

Chapter 4. Network characteristics for high availability 39

Network considerations	39
----------------------------------	----

General recommendations	40
-----------------------------------	----

DB2 connection failover recovery mechanism	45
--	----

OSPF protocol as a recovery mechanism.	45
--	----

Virtual IP Address (VIPA) as a recovery mechanism	46
---	----

Recommended setup for high availability connections between client and server	48
---	----

Alternative network setup	54
-------------------------------------	----

z/OS VIPA usage for the high availability solution for SAP.	58
---	----

Considerations when running the AS to DB and SCS connections over a private subnet	59
--	----

Timeout behavior of the client/server connection over TCP/IP	59
--	----

Timeout behavior of the AIX application server	59
--	----

Timeout behavior of the Linux application server	62
--	----

Timeout behavior of the Windows application server	63
--	----

Timeout behavior of DB2 client connections	64
--	----

Timeout behavior of the database server.	65
--	----

SAP maximum transaction time	67
--	----

Chapter 5. Concepts for a high availability SAP solution 69

Prerequisites and planning	69
--------------------------------------	----

Architecture components	69
-----------------------------------	----

SAP Central Services	70
--------------------------------	----

Network	74
-------------------	----

File system	78
-----------------------	----

Database	81
--------------------	----

Application design	83
------------------------------	----

Failure scenarios and impact.	83
---------------------------------------	----

Chapter 6. Preparing a high availability SAP solution 89

Software prerequisites	90
----------------------------------	----

Naming conventions	91
------------------------------	----

Naming conventions for Tivoli System	
--------------------------------------	--

Automation for z/OS	92
-------------------------------	----

Naming conventions used in the SA z/OS policy	94
---	----

Naming conventions for Tivoli System	
--------------------------------------	--

Automation for Multiplatforms.	95
--	----

Setting up DB2	95
--------------------------	----

Setting up file systems under z/OS for local and NFS exported file systems	95
--	----

File systems	96
------------------------	----

Setting up zFS file systems	97
---------------------------------------	----

SAP directory definitions	99
-------------------------------------	----

NFS server on z/OS	101
------------------------------	-----

NFS clients general information	103
---	-----

Non z/OS NFS server	104
-------------------------------	-----

Configuring Sysplex performance monitoring highly available with the RMF Distributed Data Server (RMF DDS).	104
---	-----

Modifying the environment for AIX and for Linux on z Systems	106
--	-----

Modifying the environment for AIX	106
---	-----

Modifying the environment for Linux on z Systems	107
Recommendations for sysplex failure management (SFM) policy definitions	109
Setting up Tivoli System Automation	109
SAP installation planning considerations	109
SAP licenses	110
SAP logon groups	110

Chapter 7. Customizing SAP for high availability. 113

General installation recommendations	114
Installation sequence	114
Setting up ABAP SCS, Java SCS, and ERS instances 116	
Installing SAP Central Services and enqueue replication server on z/OS	117
Installing the enqueue replication server as an ERS instance.	118
Installing Central Services and its enqueue replication server with ZSCSinst	118
Additional mandatory manual modifications after ASCS installation	120
Verifying ABAP SCS with enqueue replication	122
Additional mandatory manual modifications after Java SCS installation	123
Verifying Java SCS with enqueue replication .	124
SAP profile parameters	126
Installing SAP primary and additional application server instances	128
Installing the SAP primary application server instance	129
Installing additional SAP application server instances (Java-only)	129
Performing SAP post installation steps for high availability	130
The SAP Web Dispatcher	131
Installing the SAP Web Dispatcher	131
SA z/OS policy for the SAP Web Dispatcher	132
The Solution Manager Diagnostics Agent (SMDA) 133	
Installing the SMDA instance	134
Deleting an SMDA instance	135
Preparing SAP on z/OS for automation	135
Preparing the C-shell and logon profiles . . .	136
ABAP SAP Central Services (ASCS)	136
Java Central Services (SCS)	137
ABAP application server instances	137
Java and dual-stack application server instances	141
SAP host agent	142
SAProuter	143
Summary of start, stop, and monitoring commands	143

Chapter 8. Customizing Tivoli System Automation for z/OS 145

Preparing SA z/OS for SAP high availability . . .	146
How to send UNIX messages to the z/OS syslog	147
RPCBIND and NFS server - SA z/OS relationships in network configurations with and without OMPROUTE	148

Policy changes for TCPIP and OMPROUTE . . .	148
Policy changes for EnqCF replication	149
How to adapt the SA z/OS *SAPSRV <i>add-on</i> <i>policy</i>	149
Overview of the add-on policy for SAP.	150
Resource naming conventions	151
Group structure	152
Class structure	152
SAP infrastructure group	155
SAP system-dependent groups	158
ABAP central services and enqueue replication server	159
Dependencies between the ABAP enqueue server and the enqueue replication server . . .	162
Optional component of the ABAP central services	164
Java central services and enqueue replication server	164
Monitoring the health state of SAP enqueue replication	166
SAP HA Interface for SA z/OS	168
DB2 policy	169
SAP application servers as proxy resources . . .	171
Creating a remote application server policy . .	172
Scripts for the remote application server policy	174
Details and configuration for application servers as proxy resources.	174
Groups, applications, and relationships.	180

Chapter 9. Customizing System Automation for Multiplatforms 185

System Automation for Multiplatforms for SAP on IBM z Systems	185
System Automation for Multiplatforms with HA Option 2	185
How to implement the SA MP policy for SAP . . .	186
Using the SAP high availability policy feature to generate a policy	186
Using the sample SAP HA policy to generate a policy	187
Platform-specific recommendations	187
CPUPLUGD utility and Tivoli System Automation for Multiplatforms under Linux on z Systems	187
RSCT network performance considerations . .	188
Special considerations for running System Automation for Multiplatforms under virtual operating systems or LPARs	188

Chapter 10. Verifying your implementation on z/OS. 193

Verification procedures and failover scenarios . .	193
Overview of the test scenarios.	193
Test methodology	197
Planned outage test scenarios	206
Unplanned outage test scenarios	216
Problem determination methodology	231
SA z/OS problem determination	231
Where to check for application problems . . .	234
Checking the network	236

Checking the status of zFS file systems and NFS	238
Checking the status of DB2 and SAP connections	239

Chapter 11. Verifying your System Automation for Multiplatforms implementation on Linux or AIX . . . 241

Chapter 12. Operating an SAP system under System Automation control . . . 243

Managing an SAP system	243
Stopping an SAP system with SA z/OS	243
Starting an SAP system with SA z/OS	245
Stopping an SAP system with System Automation for Multiplatforms and SA z/OS	246
Checking the replication status	248
Change management during SAP operation	249
Updating DB2 or z/OS	249
Switching database connections for a single SAP application server	251
Switching database connections for multiple SAP ABAP application servers	261
Change management on z/OS UNIX	264
Change management on SAP application servers	265
Change management on SAP Central Services hosts	267
SAP tools and restarting SAP application server instances	268
Maintenance that requires a reboot of an SAP application server	268
SAP Software Update Manager (SUM)	270
Updating the SAP kernel	271
Updating the SAP kernel using a disruptive procedure	272
Updating the SAP kernel using the rolling kernel switch	272
Running the rolling kernel switch while automation is active	274

Chapter 13. Enqueue replication into a z Systems coupling facility . . . 277

High availability considerations for EnqCF replication	278
--	-----

Tool support for EnqCF replication	281
Troubleshooting for replication problems	282

Chapter 14. Reference of the z/OS high availability scripts . . . 285

Automation scripts	285
start_as	287
stop_as	287
check_as	288
start_cs	288
start_sapsrv	289
checkwd	290
Sample scripts	290

Chapter 15. Sample network setup and miscellaneous migration considerations . . . 293

Network setup	293
Network hardware components for the test setup	293
Network software components for the test setup	294
NFS client automount samples	301
Migration considerations	302
NFSv4 migration hints and tips	302
SAP System Automation policy migrations	311
Linux migration hints	313

Chapter 16. Bibliography . . . 317

Edition history of this publication and supported component versions	317
IBM documents	318
SAP documents	319
SAP Notes	319
APARs	321

Notices . . . 323

Trademarks and service marks	323
------------------------------	-----

Glossary . . . 325

Index . . . 331

Figures

1.	The closed loop of automation	5	31.	Group SAPSID_X belonging to SAP system SID	159
2.	z Systems Parallel Sysplex architecture elements	14	32.	Lowest level in group structure of ABAP central services and enqueue replication server	160
3.	DB2 data sharing in a Parallel Sysplex	15	33.	Lowest level in group structure of Java central services and enqueue replication groups	166
4.	Differences between SAP failover and CLI failover	17	34.	DB2 <i>Best Practice Policy</i> – adapted for SAP system HA1	170
5.	SAP DB2 connection failover configuration: Active/Passive example	19	35.	Server group for remote application servers	173
6.	Option 1: Automation using SA z/OS only	25	36.	Overview of the relationships for remote application servers between elements of the <i>*SAPSRV add-on policy</i>	173
7.	Option 2: Automation with the support of SA z/OS and Tivoli System Automation for Multiplatforms	27	37.	Application Symbols	177
8.	Example of high availability with GDPS configuration	36	38.	Overview of the resource groups	181
9.	Sample VSWITCH utilization	44	39.	Overview of the relationships between elements of the <i>*SAPSRV add-on policy</i> (excluding DB2 elements)	182
10.	VIPA and OSPF recovery mechanisms under z/OS.	47	40.	SM12 primary window	201
11.	Recommended setup for a high availability network.	52	41.	Error handling menu	202
12.	Alternative high availability network configuration	55	42.	Enqueue test: start mass enqueue operations	202
13.	Initial startup of SCS	72	43.	List of entries in the enqueue table	203
14.	Failure of SCS and recovery of the enqueue table	72	44.	Create Java locks	204
15.	Movement of the enqueue replication server	73	45.	List of Java locks	204
16.	Startup of SCS during failover and recovery using EnqCF replication	73	46.	SAP system log (SM21)	219
17.	Failure of SCS and recovery of the enqueue table using EnqCF replication	74	47.	Results of SDSF DA command	239
18.	General concept of a fault-tolerant network with dynamic routing and four subnets	75	48.	Results of DB2 Display Thread command	240
19.	Alternative paths in a duplicated network	76	49.	NetView command session display for stopping SAP application servers (1)	244
20.	Rerouting if a network adapter card fails	76	50.	INGLIST panel where all SAP application server resources are stopped	245
21.	VIPA takeover and dynamic routing	77	51.	DB connection list for the SAP application server on host ih1scohlv.	256
22.	Initial NFS client/server configuration	81	52.	Active DB connections	256
23.	Failover of the NFS server	81	53.	Data sharing topology	257
24.	High availability solution configuration for SAP	90	54.	Select the target connection name and switch database connection	258
25.	SAP directory structure and file systems	97	55.	ABAP work processes information in the Data Sharing Topology view	258
26.	SA z/OS <i>*SAPSRV add-on policy</i> for SAP	152	56.	Select the target connection name and switch database connection	259
27.	<i>*SAPSRV add-on policy</i> - class structure for SAP Central Services resources.	154	57.	Double-click the target DB connection	261
28.	<i>*SAPSRV add-on policy</i> - class structure for HA1 sample system	155	58.	Networking configuration for the high availability solution for SAP	293
29.	SAP-system-independent groups and resources	156			
30.	<i>*SAPSRV add-on policy</i> - class structure for Solution Manager Diagnostics Agent resources	158			

Tables

1. Parallel Sysplex availability features matrix	12	24. Restart the SAP infrastructure group	210
2. Recovery attributes of the recommended setup	52	25. Stop of all SAP HA2 application servers with SA z/OS	211
3. Retransmission intervals	61	26. Start of all SAP HA2 components with SA z/OS	211
4. High availability configurations	84	27. Shutdown of the LPAR where the ES and NFS servers are running	213
5. Failure scenarios and impact	85	28. Restart of the LPAR where the ES and NFS servers were running previously	213
6. Software requirements for the HA solution	90	29. Startup of the first LPAR	215
7. Software requirements for SAP application servers	91	30. Startup of the remaining LPARs	216
8. Recommended names for all z/OS-related components of an SAP system	93	31. Failure of the SAP enqueue server	217
9. Recommended names for all components of an individual SAP system	93	32. Failure of the message server	220
10. Naming conventions for SA z/OS resources	94	33. Failure of the enqueue replication server	221
11. Enqueue client parameters relevant for the high availability solution	126	34. Failure of the SAP start service	222
12. Enqueue server parameters relevant for the high availability solution	127	35. Failure of the sapstart process of the SAP Central Services	224
13. Summary of start/stop monitoring commands	143	36. Failure of the sapstart process of the enqueue replication server	225
14. Differences between entry names and subsystem or automation names	151	37. Failure of the NFS server	226
15. SID-specific classes	153	38. Failure of a TCP/IP stack	227
16. Messages and User Data section from the SAPSIDACV policy definition	163	39. Failure of the LPAR where the ES and NFS servers are running	229
17. Startup section from the SAPSIDAER policy definition	163	40. Comparison of <i>EnqCF replication</i> and <i>TCP/IP-based replication</i>	278
18. Relationships section from the SAPSIDAER policy definition	163	41. Scripts for the <i>*SAPSRV add-on policy</i> of SA z/OS	286
19. Examples of test scenarios	194	42. Debugging and test commands	309
20. Automatic restart of sapstartsrv after patch level upgrade	206	43. Edition history of Business Continuity for SAP on IBM z Systems	317
21. Stop of all SAP HA2 components with SA z/OS	207	44. Other IBM reference documents	318
22. Start of all SAP HA2 components with SA z/OS	207	45. IBM Redbooks and Redpapers covering related topics	318
23. Stop the SAP infrastructure group	210	46. SAP documents	319
		47. Relevant SAP Notes	320

Summary of changes in edition 12/2016

This edition of the document describes technical enhancements of the IBM® *High Availability Solution for SAP on z Systems*™. All technical changes or terminology, maintenance, and editorial changes to the text are indicated by a vertical line to the left of the change.

- This edition is based on System Automation for z/OS 3.5 with APAR level OA51440 .
- The topic Chapter 3, “Planning for high availability for SAP,” on page 21 has been updated, it now recommends two high availability options:
 - “Option 1: Automation using System Automation for z/OS only” on page 24, which is the preferred choice, and the alternative option:
 - “Option 2: Automation by System Automation products with central components on AIX or Linux” on page 26

For details see “How to decide which System Automation option to implement” on page 24.

- A new description of the rolling kernel switch procedure in HA environments is described in “Running the rolling kernel switch while automation is active” on page 274.
- The description of the DB2® connection failover focuses on CLI failover (see “DB2 connection failover” on page 15). The previous SAP failover option is no longer supported with DB2 12 and DB2 Connect™ version 11.1.

Summary of changes in edition 12/2015

System Automation for z/OS 3.5 with APAR level OA48922 provides the following enhancements:

- The shutdown procedure for SA z/OS remote application server resources is changed. For details see “Java and dual-stack application server instances” on page 141.
- The SA z/OS **SAPSRV add-on policy* no longer models an optional SAP gateway service for ABAP SAP Central Services (ASCS).
- You can use the **SAPSRV add-on policy* to automate a Solution Manager Diagnostics Agent (SMDA) that is installed on z/OS® as a standalone engine.
- The SAP HA Interface for SA z/OS is enabled for SAP Web Dispatcher and SAP Solution Manager Diagnostics Agent instances. (see “SAP HA Interface for SA z/OS” on page 168).

The end-to-end automation using the System Automation Application Manager has been removed and is no longer described in this Edition 12/2015. It is still available in earlier versions of this publication. With this Edition 12/2015, the recommended configuration option is the implementation that concentrates on SA z/OS described in “Option 1: Automation using System Automation for z/OS only” on page 24.

Furthermore, Tivoli® System Automation for Multiplatforms with its *SAP high availability policy feature* is now recommended for automating SAP components on AIX® or Linux. Refer to Chapter 9, “Customizing System Automation for Multiplatforms,” on page 185 and Chapter 11, “Verifying your System Automation for Multiplatforms implementation on Linux or AIX,” on page 241.

Summary of changes in edition 12/2014

The **SAPSRV add-on policy* that is part of System Automation for z/OS 3.5 with APAR OA46166 provides the following enhancements:

- It models the SAP Web Dispatcher based on the standard SAP infrastructure and the SAP start service `sapstartsrv`. So the new policy uses `sapstartsrv` to control SAP Web Dispatcher instances. It is supported by the HA-Wizard coming with the previously mentioned APAR level.
- It supports the SAP HA Interface Version 2 (see “SAP HA Interface for SA z/OS” on page 168).
- The supporting scripts of version 3.0, `start_cs`, `start_sapsrv`, `samsapctrl_asping`, and `sapolicy.sap.map_USS` are based on the new IBM cross platform script `sap_xplatform`. `sap_xplatform` is used by the IBM products SA z/OS, System Automation for Multiplatforms, and Power® HA. It contains cross platform functions to start, stop and monitor SAP resources. This cross platform integration increases the reliability of the policy to an even higher level.
- REXEC is no longer supported as communication protocol from z/OS UNIX to a remote SAP application server. This is for security reasons. SSH is the only communication protocol supported.

This edition of the document includes the following new information:

- “Configuring Sysplex performance monitoring highly available with the RMF Distributed Data Server (RMF DDS)” on page 104 is a new topic with configuration information about how to exploit the RMF™ DDS in order to obtain a highly available performance monitoring of all systems in the sysplex where your SAP environment is running. Configuration information for the DBA Cockpit in SAP is also included.
- DBA cockpit supports the DB2 CLI driver based failover. Therefore, “DB2 connection failover for ABAP instances” on page 16 has been updated with precised information about DB2 CLI failover. Also, Chapter 12, “Operating an SAP system under System Automation control,” on page 243 now describes the DB2 (or z/OS) update on the basis of the DB2 CLI based failover.
- “Updating DB2 or z/OS” on page 249 contains information about how switching multiple ABAP application server connections is simplified with report `RSDB2SWITCH` for SAP installations with SAP failover.

About this document

This publication describes a solution for *Business Continuity for SAP on IBM z Systems*, which provides the means for fully automating the management of all SAP components and related products running on z/OS, AIX, Linux, or Windows. The automation software monitors all resources and controls the restart, or takeover of failing components, or both, thereby ensuring a high and almost continuous availability of the SAP system. Therefore, emphasis in this document is put on solutions for high availability as this is the most important aspect of business continuity, besides the availability of DB2 data.

Who should read this document

This document is intended for system and database administrators who need to support SAP systems that must offer a high level of availability.

Prerequisite and related information

SAP on DB2 uses a variety of different hardware and software systems. This document concentrates on information that goes beyond the standard knowledge needed for DB2 and SAP system administration. Therefore, it is assumed that you are familiar with:

- The z/OS environment (TSO, UNIX System Services, RACF®, JCL, RMF, WLM)
- DB2 administration (for example, SQL, SPUFI, and the utilities REORG and RUNSTATS)
- AIX or Linux on z Systems (depending on your choice of SAP application servers)
- Linux on System x
- Windows

Refer to Chapter 16, “Bibliography,” on page 317 for a list of related documentation.

Additional information is available from SAP as part of the help system:
<http://help.sap.com>

and on the SAP Community Network (SCN) web site for SAP on DB2 for z/OS:
<https://go.sap.com/community/topic/db2-for-zos.html>

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this document or any other z/OS documentation:

- Visit the **SAP on IBM z Systems Community** at:
<http://ibm.co/1ToJ1R8>
- Use the Discussion-Forum for your questions.
- Fill out one of the forms at the back of this document and return it by mail, by fax, or by giving it to an IBM representative.

Content of this document

This document describes concepts and activities which are necessary to set up an *SAP on DB2 for z/OS* system which runs highly available as part of a Business Continuity solution based on the System Automation product family and GDPS®.

Chapter 1, “Introduction of high availability and automation for SAP,” on page 1 through Chapter 3, “Planning for high availability for SAP,” on page 21

This part of the document provides general information and planning considerations for high availability in an SAP environment.

Chapter 4, “Network characteristics for high availability,” on page 39

This information unit describes a highly-available network established for testing and makes general recommendations concerning network setup. It also discusses the implementation of a high availability solution as it affects the client/server configuration and addresses timeout considerations.

Chapter 5, “Concepts for a high availability SAP solution,” on page 69 through Chapter 7, “Customizing SAP for high availability,” on page 113

This part of the document discusses the components of the architecture, including considerations for SAP Central Services (SCS), network, file system, database, information on remote application servers and DB2 connection failover. It offers scenarios showing different high availability implementations and also gives information on planning for high availability implementation, with considerations for DB2, network, file system, Tivoli System Automation, and SAP installation. Finally, it describes what is needed to adapt the SAP system to the high availability solution, including configuring SAP for SCS and for Tivoli System Automation.

Chapter 8, “Customizing Tivoli System Automation for z/OS,” on page 145 and Chapter 9, “Customizing System Automation for Multiplatforms,” on page 185

This part of the document discusses the customization of Tivoli System Automation products, namely System Automation for z/OS and System Automation for Multiplatforms. It also discusses change management strategies when you update and upgrade system components.

Chapter 10, “Verifying your implementation on z/OS,” on page 193 and Chapter 11, “Verifying your System Automation for Multiplatforms implementation on Linux or AIX,” on page 241

This part of the document contains procedures and checklists which allow you to verify that the high-availability implementation and the customization of the Tivoli System Automation products is correct.

Chapter 12, “Operating an SAP system under System Automation control,” on page 243

This information unit describes how to perform daily tasks on and manage changes to your SAP system with System Automation. It also contains information on how check the replication status to determine whether the enqueue table is completely replicated after a replication server restart.

Chapter 13, “Enqueue replication into a z Systems coupling facility,” on page 277

Besides the standard SAP enqueue replication, an additional replication mechanism is available for SAP on IBM z Systems. This information describes the business continuity aspects of this additional replication mechanism, and how it can be integrated into an existing Business Continuity setup for SAP.

Chapter 14, “Reference of the z/OS high availability scripts,” on page 285

This information unit describes all automation scripts used in the current edition of this publication. For each script, its purpose, the invocation syntax, and an explanation of the parameters is provided. Also, you can read how to obtain the scripts.

Chapter 15, “Sample network setup and miscellaneous migration considerations,” on page 293

This topic contains miscellaneous information in two subtopics. The first subtopic outlines a highly available network that was part of a test implementation of a business continuity solution for SAP on DB2. The second subtopic presents migration hints and tips on NFSv4 migration and SAP System Automation policy migration, and also presents helpful hints and tips on Linux for SLES 11 and SLES 12.

Chapter 1. Introduction of high availability and automation for SAP

Business continuity of an SAP production system is a critical business factor. It requires the highest possible level of system availability. The solution is described in this document.

This high availability solution for SAP offers multiple advantages.

- It combines high availability techniques with automation technologies of IBM Tivoli System Automation products.
- It helps to avoid unplanned outages by eliminating single points of failure.
- It helps to avoid planned outages such as administrative or maintenance work.
- It provides business continuity for an SAP production system as close as possible to 24 hours a day during 365 days a year.

IBM Systems products incorporate various advanced autonomic computing capabilities that are based on the four characteristics of self-managing systems:

Self-configuring

The seamless integration of new hardware resources and the cooperative yielding of resources by the operating system is an important element of self-configuring systems. Hardware subsystems and resources can configure and reconfigure autonomously both at startup time and during run time. Based on the current optimization criteria, or in response to hardware or firmware faults, this action can be initiated by the need to adjust the allocation of resources. Self-configuring also includes the ability to concurrently add or remove hardware resources in response to commands from administrators, service personnel, or hardware resource management software.

Self-healing

With self-healing capabilities, operating systems can detect hardware and firmware faults instantly and then limit the effects of the faults within defined boundaries. These capabilities allow systems to recover from the negative effects of such faults with minimal or no impact on the execution of operating system and user-level workloads.

Self-optimizing

Self-optimizing capabilities allow computing systems to autonomously measure the performance or usage of resources and then tune the configuration of hardware resources to deliver improved performance.

Self-protecting

Self-protecting capabilities allow computing systems to protect against internal and external threats to the integrity and privacy of applications and data.

Since the announcement of SAP on DB2 for z/OS, DB2 Parallel Sysplex® data sharing combined with *DB2 connection failover* has been used to remove the database server as a single point of failure. These features can help you to avoid planned and unplanned outages of the database server.

The high availability solution, which is presented in this document, further enhances business continuity by removing the SAP central instance as a single

point of failure. This solution also provides a method to automate the management of all SAP components for planned and unplanned outages. Thus, high availability is achieved by combining the concepts of system automation and transparent failover in a Parallel Sysplex. Based on the IBM Tivoli System Automation products, together with a redesign of the SAP central instance concept, this high availability solution exploits the SAP stand-alone enqueue server, the enqueue replication server, dynamic virtual IP addresses (VIPA), shared file system, and DB2 data sharing to aim for a minimum of SAP system outages together with maximum automation.

The implementation and customization of the complete high availability solution highly depends on the customer configuration and requires IBM Tivoli System Automation skill. It is recommended that customers request support from IBM Global Services. Before customers go into production with their implementation of the solution, they should also contact SAP for a final check of the setup.

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

The availability of the enqueue server is critical for an SAP system. If it fails, most SAP transactions also fail. To address this single point of failure, SAP, in cooperation with IBM, changed the architecture of the enqueue server. It is no longer part of the so-called *central instance*. That is, it no longer runs inside a work process, but is now a stand-alone process, which is called the *stand-alone enqueue server*. It operates under the designation *SAP Central Services*, or SCS. The enqueue server transmits its replication data to an enqueue replication server, which runs on a different system. The enqueue replication server stores the replication data in a shadow enqueue table in shared memory. The SAP developer network (SDN) provides more information about this topic and about high availability. For more information about the SAP enqueue server and replication server, see “SAP Central Services” on page 70. Also, refer to the description of the SAP high availability architecture and the SAP Lock Concept, which can be found on the SAP NetWeaver documentation:

SAP Lock Concept

If the enqueue server fails, it is quickly restarted by Tivoli System Automation on the system where the replication server was running. It uses the replicated data in the shadow enqueue table to rebuild the tables and data structures. Thus, a failure of the enqueue server is not visible to the user and the SAP application. For a detailed description of this process, see Chapter 5, “Concepts for a high availability SAP solution,” on page 69.

The *business continuity solution*, which is described in this document, is derived from an SAP test environment that is the blueprint of implementing an almost continuously available SAP system on DB2 for z/OS.

The IBM product Tivoli System Automation was chosen for the *business continuity solution*, because it not only provides the means for the implementation of a high availability system, but also includes all features needed to streamline daily operations. For example, it includes features for automated start-up, shutdown, and monitoring of the components of an SAP system and its dependent products.

Because of these capabilities, System Automation is also a prerequisite for Geographically Dispersed Parallel Sysplex™ (GDPS). See “GDPS infrastructure for disaster recovery” on page 34.

High availability definitions

Certain terms are used to indicate various degrees of availability and two types of outages that affect availability.

- “Degrees of availability”
- “Types of outages”

Degrees of availability

The terms *high availability*, *continuous operation*, and *continuous availability* are used to express how available a system is.

High availability

High availability refers to the ability to avoid unplanned outages by eliminating single points of failure. The absence of unplanned outages 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 hiding the outage from the users. This hiding can be accomplished by quickly restarting failed components with the help of an automation program such as IBM Tivoli System Automation for z/OS.

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 users. This continuous operation is accomplished by providing multiple servers and switching users to an available server at times when one server is made unavailable. Using DB2 data sharing with DB2 connection failover is an example of how continuity is accomplished in an SAP environment. Topics Chapter 2, “SAP availability benefits provided by IBM z Systems,” on page 11 through “Disaster recovery” on page 32 describe how a number of planned outages can be avoided by taking advantage of DB2 data sharing and DB2 connection failover.

Note: A system that is running in continuous operation is not necessarily operating with high availability because an excessive number of unplanned outages can compromise availability.

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 24 hours a day during 365 days a year as possible.

Types of outages

Availability of the SAP system is a critical business factor, and the highest level of availability must be provided. Therefore, you must be aware of the types of outages (planned and unplanned) and how to avoid them.

Planned outage

Planned outages are deliberate and are scheduled at a convenient time, like for example:

- 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 unexpectedly caused by the failure of any SAP system component. They include hardware failures, software problems, or people and process issues.

About one-fifth of unplanned outages result from application or other software errors. These include software failures, application changes, or performance issues.

In addition, about one-fifth of unplanned outages result from operator errors and unexpected user behavior. These include changes to system components, not executing tasks or executing tasks incorrectly 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.

The autonomic computing self-healing technologies of Tivoli System Automation

To avoid all causes of outages, the high availability solution uses the autonomic computing self-healing technologies that are implemented in Tivoli System Automation. Tivoli System Automation can automatically discover system, application, and resource failures in a cluster.

It uses sophisticated, policy-based knowledge about application components and their relationships, and also uses their availability goals to decide on corrective actions within the correct context. Tivoli System Automation manages the availability of business applications, which are running on single systems and clusters on z/OS and Linux on z Systems (and others). Tivoli System Automation for z/OS plays an important role in building the end-to-end automation of the IBM autonomic computing initiative. Its unique functions are designed to automate system operations (and I/O and processor) in a closed loop as shown in Figure 1 on page 5.

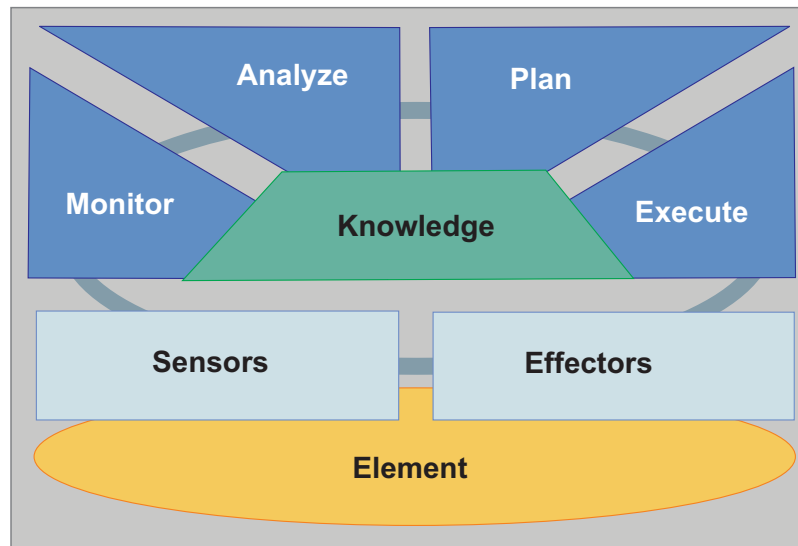


Figure 1. The closed loop of automation

Resource elements are monitored by sensors. The automation engine analyzes the current status and compares it with the goal status of the resource. If the current status and goal status differ, then the automation engine uses the policy to deduce a plan to bring the resource and entire system into the wanted state. The plan is processed via effectors to the resource element, and the loop then starts again.

This process is known as *policy-based self-healing*.

High availability and automation objectives for SAP

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

The following methods are used for this purpose:

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

No planned outages

Planned outages for software or hardware maintenance can be avoided by using *Parallel Sysplex data sharing* and *DB2 connection failover* to dynamically move application server instances to standby database servers. It is possible to have multiple standby database servers that allow a cascade of moves. More information about multiple standby database servers is documented in topics Chapter 2, “SAP availability benefits provided by IBM z Systems,” on page 11 through “Disaster recovery” on page 32.

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

If SAP Central Services (SCS) are running on the system where maintenance is to be applied, system automation can be used to move SCS to a standby z/OS LPAR. This move is not apparent to the users. SAP work processes automatically reconnect to the moved SCS without failing any transactions.

Reduced operator errors

The high availability solution for SAP uses IBM Tivoli System Automation to automate the starting, stopping, and monitoring of all SAP components. By automating daily operations, there is less opportunity for error during the process of starting or stopping SAP components. With Tivoli System Automation, you can define component dependencies with parent-child relationships. In doing this, SA checks that a component that has a parent is not started before its parent is active. Tivoli System Automation also checks that a component is not stopped if there are active child components. This ensures that an orderly start or stop of the SAP system is accomplished with little opportunity for operator error. See Chapter 7, “Customizing SAP for high availability,” on page 113 for a description of the setup, and see Chapter 12, “Operating an SAP system under System Automation control,” on page 243 for a description of common operations scenarios.

System Automation for Multiplatforms automates only those SAP components running outside of z/OS.

Health check for application problems

To facilitate ABAP application server monitoring, IBM provides the utility `samsapctrl_asping` (see “ABAP application server instances” on page 137).

The `samsapctrl_asping` utility is also used in the SAP policy for SAP Java™ application server monitoring. This utility keeps running as long as the `sapcontrol` function is successful. If `sapcontrol` returns that the instance is stopped, `samsapctrl_asping` stops. This stopping signals to Tivoli System Automation that the application server instance is down.

Overview of the high availability solution for SAP

Read this information to learn about solutions for high availability for SAP systems and SAP system automation.

High availability of an SAP system

As described in “Degrees of availability” on page 3, single points of failure must be eliminated. DB2 data sharing is used to remove the database server as a single point of failure. With SCS, the enqueue server was removed as a single point of failure. The high availability solution for SAP also adds a movable Network File System server (NFS server) and dynamic virtual IP addressing (under z/OS only) so that application components can be moved. IBM Tivoli System Automation is used to monitor these components and quickly restart them if they fail.

Automating an SAP system

The high availability solution for SAP uses IBM Tivoli System Automation for z/OS to automate all SAP components.

- DB2 subsystems
- enqueue server

- message server
- enqueue replication server (if required)
- TCP/IP
- NFS server
- the SAP Java gateway server
- Optionally:
 - the SAP Web Dispatcher
 - the SAP Solution Manager Diagnostics Agent (SMDA)

The optional components, SAP syslog sender, and collector, which existed in earlier SAP NetWeaver releases, are now obsolete. Refer to *SAP Note 1041390: SM21: Central system log via HTTP or HTTPS* for more details.

By automating all the SAP components, the SAP system can be started, stopped, and monitored as a single resource. This automation provides for the highest level of availability by reducing operator commands, and thus reducing operator errors (see also Chapter 12, “Operating an SAP system under System Automation control,” on page 243).

System Automation for Multiplatforms can automate those SAP components, which run outside of z/OS, that is SAP ABAP or Java application servers under Linux or AIX. Additionally, if you decided to run SAP Central Service or the NFS server outside of z/OS, under Linux or AIX (as described in “Option 2: Automation by System Automation products with central components on AIX or Linux” on page 26), System Automation for Multiplatforms also automates them.

The SAP on z Systems solution is inherently heterogeneous in that it always involves z/OS components and at least one more operating system on which the SAP application servers run.

Benefits of IBM Tivoli System Automation

Because an SAP system has many components that operate in a complex manner, 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.

IBM Tivoli System Automation (SA) offers system-wide benefits by simplifying the operation of the entire SAP system, which is 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 correct order. Failure to observe the correct order delays the system's availability.

In IBM Tivoli System Automation, the emphasis is on goal-driven automation. Automation programmers 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 IBM Tivoli System Automation provides both the speed and a high degree of automation. On the other hand, it avoids the complexity of scripted automation tools, thus reducing automation errors.

The automation manager works to keep systems in line with these goals. It 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 required checks and decisions can be high and a human cannot perform the same tasks as fast and reliably as the automation manager.

Goal-driven automation simplifies operations. Operators request what they want, and automation takes care of any dependencies and resolves affected or 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.

You can find more information about IBM Tivoli System Automation on the web:

- For Tivoli System Automation for z/OS at:
<http://www.ibm.com/software/products/en/tivosystautoforzos>
- For Tivoli System Automation for Multiplatforms at:
<http://www.ibm.com/software/products/en/tivosystautoformult>

Conventions and terminology used in this document

Read this information to learn about the naming conventions that apply to this publication.

- The abbreviation SID is used for the SAP system name (or SAP system identification). When this SID occurs as a placeholder for a real SAP system name (for example, as part of resource names in the examples of this documentation), then this is denoted in italics in the figures and additionally within angled brackets in the text. For example, if the SAP system name is WD1, the notation SAP<SID>WD_ST can stand for the SAP Web Dispatcher sapstart resource SAPWD1WD_ST.

The SAP administrator's name is denoted with <sid>adm.

In some cases, the abbreviation SAPSID is used as a synonym for SID.

- IBM DB2 for z/OS is referred to as **DB2**.
- The *SAP on DB2 for z/OS* system is referred to as **SAP on DB2**.
- The term **UNIX** stands for AIX and z/OS UNIX System Services. *UNIX(-like)* or *UNIX(-style)* refers to *UNIX* and *Linux*.
- AIX 6.x or AIX 7.x are referred to as **AIX**.
- Linux on z Systems (64-bit) and Linux on System x (64-bit) are both referred to as *Linux*.
- The term Windows is used to encompass Windows Server 2008 R2 and its supported successors.
- The IBM products Tivoli System Automation for z/OS (SA z/OS) and Tivoli System Automation for Multiplatforms are both referred to as *Tivoli System Automation (SA)* in this document.
- IBM Tivoli System Automation for Multiplatforms is referred to as System Automation for Multiplatforms.
- The term NetView[®] refers to the IBM product Tivoli NetView for z/OS.

- DB2 documentation is cited without a specific release or order number, because these numbers are different for different DB2 versions. Refer to Chapter 16, “Bibliography,” on page 317 for specific information.
- *Planning Guide* refers to the *Planning Guide for SAP on IBM DB2 for z/OS*, which is available for the different SAP 7.x NetWeaver releases. Refer to the SAP Community Network (SCN) web site for SAP on DB2 for z/OS:
<https://go.sap.com/community/topic/db2-for-zos.html>
- The SAP documentation that is specific to the database implementation is referred to as the *Database Administration Guide: SAP on IBM DB2 for z/OS* (see “SAP documents” on page 319 for full titles). Do not confuse this with the *IBM DB2 Administration Guide* publication.
- The term *SAP Installation Guide* refers to the release-specific SAP installation documentation (“SAP documents” on page 319).
- SAP designated SAP Central Services for ABAP as ASCS (ABAP SAP Central Services) and now applies the abbreviation SCS to the Java-based variant. This terminology is attributable to the use of these abbreviations as directory names. However, this publication continues to use the abbreviation SCS as a conceptual term and to refer to an SCS instance in general. It employs the terms ASCS and *Java SCS* to designate the environment-dependent instances when required. See “SAP Central Services” on page 70.

Chapter 2. SAP availability benefits provided by IBM z Systems

The IBM z Systems platform incorporates various advanced availability computing capabilities.

The computing capabilities of IBM z Systems include hardware features as well as features of the involved software components. Thus, IBM z Systems provide an integrated infrastructure for a highest possible availability for the SAP solution of an enterprise.

The goal of this infrastructure is to eliminate any possible single point of failure through redundancy, on both the hardware and software sides. Furthermore, when a failure occurs, the system should record sufficient information about it so that the problem can be fixed before it recurs. For software, the purpose of this written information is not only to avoid failures but also to identify and recover those failures that occur. Automation eliminates failures by ensuring that procedures are followed accurately and quickly every time.

The availability features of the z Systems platform are derived from these concepts. z Systems was designed with the reliability, availability, and serviceability (RAS) philosophy. Its availability features result from 50 years of evolution and are incorporated in the z Systems hardware, the z/OS operating system, and DB2 for z/OS.

For details of the current z Systems hardware architecture and its high availability features that are built into z Systems servers, refer to the website

<http://www-03.ibm.com/systems/z/hardware/zenterprise/>

Features of z/OS

Read this topic for information about the approach of the z/OS operating system to support availability of systems and applications.

The z/OS operating system has a reliability philosophy that recognizes the inevitability of errors. This philosophy dictates a comprehensive approach to error isolation, identification, and recovery rather than a simplistic automatic restart approach. In support of this comprehensive approach, z/OS provides a vast array of software reliability and availability features, far beyond those features currently provided by any other operating system. A large portion of the z/OS kernel operating system exists solely to provide advanced reliability, availability, and serviceability capabilities. For example, here are some RAS guidelines that must be obeyed:

- All code must be covered by a recovery routine, including the code of recovery routines themselves. Multiple layers of recovery are therefore supported.
- All control areas and queues must be verified before processing continues.
- Recovery and retry must be attempted if there is hope of success.
- All failures that cannot be transparently recovered must be isolated to the smallest possible unit, for example the current request, a single task, or a single address space.

Diagnostic data must be provided. Its objective is to allow the problem to be identified and fixed after a single occurrence. The diagnostic data is provided even when retry is attempted and succeeds.

Note: For a detailed list and description of these features, refer to the information provided at the following link.

<http://www.ibm.com/systems/z/os/zos/features/>

Availability features and benefits with z Systems Parallel Sysplex

Parallel Sysplex technology is the basis for achieving high availability for your SAP systems.

The following link contains a complete list of advantages and benefits:

<http://www.ibm.com/systems/z/advantages/ps/index.html>

List of z Systems Parallel Sysplex availability features

The following table summarizes the features, which are implemented in the design of DB2 for z/OS. It shows which availability features apply to the frequency, duration, and scope of an outage. It further explains whether this feature helps eliminate planned or unplanned outages, or both.

Table 1. Parallel Sysplex availability features matrix

Availability feature	Reduces outage frequency	Reduces outage duration	Reduces outage scope	Planned outage	Unplanned outage
Data sharing	X	X	X	X	X
Non-disruptive hardware changes	X	X	X	X	X
Non-disruptive software changes	X	X	X	X	X
Non-disruptive policy changes	X	X	X	X	X

X = applies

- **DB2 data sharing**

The DB2 data sharing technology eliminates single points of failure by running multiple DB2 engines (so-called DB2 members) in parallel that process the same data. The z Systems Coupling Facility as central hub for the cooperation and synchronization of the DB2 members enables efficient data sharing processing. With data sharing, you can apply DB2 release upgrades and DB2 maintenance in an online fashion. For details on how to perform an online DB2 release upgrade with an SAP application, refer to the SAP Best Practices Guide: *Migrating SAP Systems to DB2 11 for z/OS*, or to DB2 12 for z/OS. They are available at:

<https://go.sap.com/community/topic/db2-for-zos.html>

DB2 data sharing is fully compatible with HyperSwap® and GDPS technologies for disaster recovery. To minimize the data sharing overhead, DB2 data sharing exploits the explicit hierarchical locking (EHL) technology.

- *Non-disruptive hardware changes:*

Capacity can be dynamically added in incremental steps: processor, LPAR, and CEC. The non-disruptive hardware changes category also covers the removal of a system member from the Parallel Sysplex.

- *Non-disruptive software changes:*

Both z/OS and DB2 for z/OS support non-disruptive software changes. Thus, individual instances of an element can be upgraded by removing that element from the sysplex and adding the upgraded element back when it is ready. Therefore, both the old and new versions must co-exist and work together within the Parallel Sysplex. For more information on this release tolerance, see “Updating DB2 or z/OS” on page 249.

- *Non-disruptive policy changes:*

The Sysplex Failure Manager is used to describe a set of actions that the Parallel Sysplex processes in the event of certain failures. These failures can range from the loss of an LPAR, where the remaining active LPARs can be enabled to automatically take the storage from the failing LPAR, to failures within database subsystems. The active set of instructions is known as a Sysplex Failure Manager Policy, and this policy can be changed dynamically without a service interruption.

DB2 data sharing on z Systems Parallel Sysplex

There are good reasons and motivations for pursuing an SAP implementation, which is based on z Systems Parallel Sysplex and DB2 data sharing.

The following topics are contained:

- “Why Parallel Sysplex and data sharing for SAP?”
- “Parallel Sysplex architecture” on page 14
- “DB2 data sharing architecture” on page 14

Why Parallel Sysplex and data sharing for SAP?

Many customers are deploying SAP applications in support of business-critical operations. Typical business drivers are characterized by the following features:

- a desire to run a single global SAP instance
- Customer and supplier access to web-based SAP applications around the clock
- Support of 24 hours a day x 365 days a year manufacturing
- Distribution operations
- Real-time core banking

These business drivers lead to the following IT requirements:

- Near-continuous system availability
For a definition of *continuous availability* and the high availability and automation objectives for SAP, see Chapter 1, “Introduction of high availability and automation for SAP,” on page 1.
- Central processor scalability through horizontal growth

The infrastructure for the high availability solution, which is described in the following, is essential for horizontal processor scalability as well. Historically systems grew *vertically* by adding engines to the machine (also known as a symmetric multiprocessor or SMP or CEC) or by introducing faster processors. This approach limited the size of an SAP system to the largest single SMP or CEC. The SAP DB2 Parallel Sysplex architecture enables users to overcome these constraints and cluster multiple CECs in a single DB2 data sharing group. This approach enables horizontal growth of both processor power (MIPS) and real memory. Data sharing also gives users another means in workload management as they can now level multiple workloads across two or more machines.

Parallel Sysplex architecture

A fundamental building block for both high availability and continuous operations is the notion of clustered database servers, which are operating against a single copy of the data. Figure 2 introduces a high-level picture of the elements of a z Systems Parallel Sysplex environment.

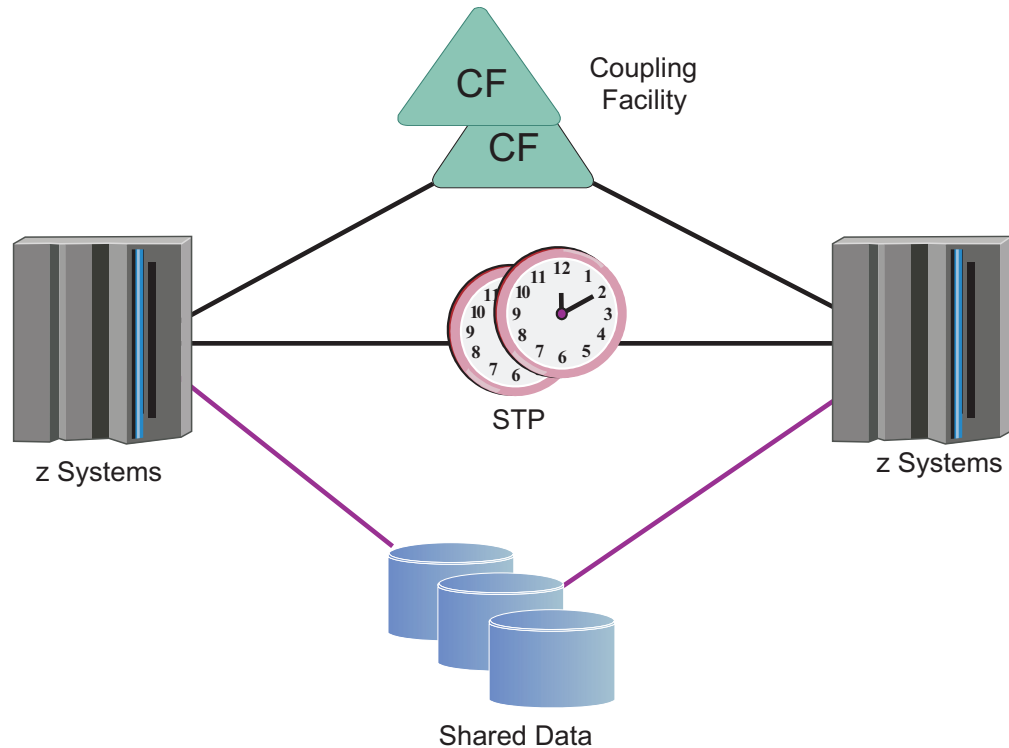


Figure 2. z Systems Parallel Sysplex architecture elements

In Figure 2, you see that a Parallel Sysplex is typically made up of the shown components:

- Two or more computer systems (known as a Central Electronic Complex or CEC)
- Two or more Coupling Facilities (either internal, ICF, or external) for the storing of shared operating system and DB2 structures between the CECs
- Two external time sources that are provided via the Server Time Protocol (STP) feature

Note: Earlier generations of servers such as IBM system z9[®] and IBM system z10[™] also provided the option to use a Sysplex Timer. IBM zEnterprise[®] 196 servers or later use only STP.

- Sysplex-wide shared data
- Multiple high-speed, duplexed links that connect the components

This implementation employs hardware, software, and microcode.

DB2 data sharing architecture

Figure 3 on page 15 completes the picture by laying multiple DB2 data sharing members on top of the Parallel Sysplex infrastructure.

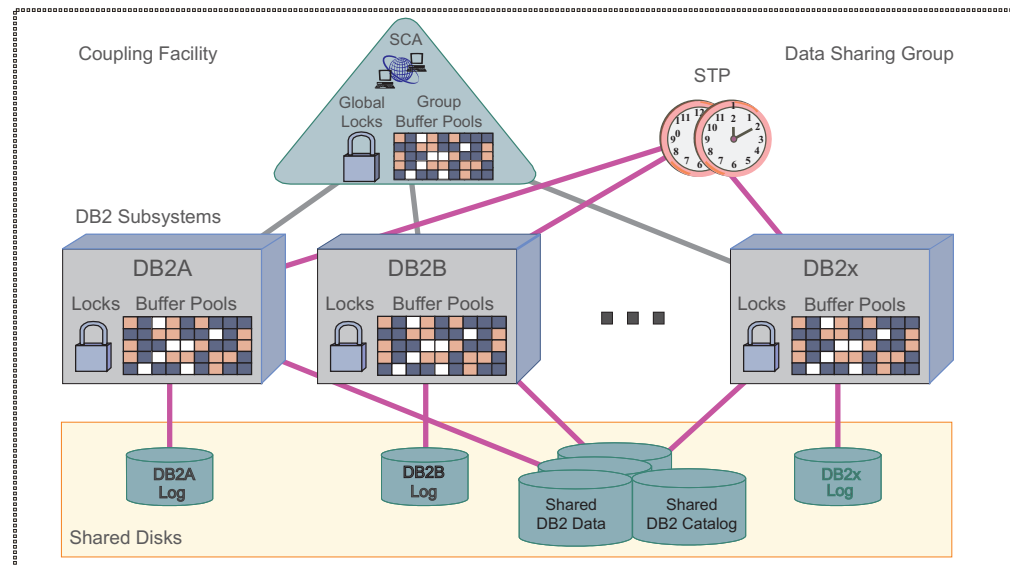


Figure 3. DB2 data sharing in a Parallel Sysplex

In Figure 3 you see up to 32 DB2 subsystems (DB2 members) making up a DB2 data sharing group. Each DB2 subsystem has its own set of DB2 logs, local buffer pools, and local locks that are managed by a companion IRLM. The DB2 data sharing group shares the SAP tables and indices, the DB2 catalog/directory, and the DB2 data sharing structures (SCA, global locks, group buffer pools) stored in the coupling facility.

Data sharing concepts for DB2 for z/OS are explained in more detail in the *IBM Knowledge Center for DB2 for z/OS*.

DB2 connection failover

Read an overview of DB2 connection failover, which should be implemented to complete the DB2 data sharing infrastructure.

With planned or unplanned outages of DB2 data sharing members, the term *DB2 connection failover* refers to the capability of an SAP system to switch its DB2 database connection to another DB2 data sharing member. Thus, a (partial or even complete) SAP system outage is avoided.

SAP application servers use the DB2 Connect product to connect to the DB2 z/OS database. In particular:

- SAP ABAP instances connect to DB2 via the *IBM Data Server Driver for ODBC and CLI*. In the following we refer to this driver as *DB2 CLI driver*.
- SAP Java instances connect to DB2 via the *IBM Data Server Driver for JDBC and SQLJ*. In the following we refer to this driver as *DB2 JDBC driver*.

For a description on how you can apply non-disruptive maintenance to your system with the help of DB2 connection failover, see “Change management during SAP operation” on page 249 and in particular “Updating DB2 or z/OS” on page 249.

The implementation of DB2 connection failover varies according to the type of the SAP instance:

- connection failover for ABAP type SAP instances, see “DB2 connection failover for ABAP instances” with two alternatives:
 - implemented as an SAP function
 - implemented with native CLI driver failover capabilities
- connection failover for Java type SAP instances using the DB2 JDBC driver failover capabilities, see “DB2 connection failover for Java instances” on page 18.

Finally, “Building blocks for DB2 connection failover” on page 19 introduces two major failover configurations that you can select for each of the three mentioned implementation flavors:

- Active/Passive: Single DB2 member with passive (inactive) standby member
- Active/Active: Two or more active DB2 members in active-standby mode

DB2 connection failover for ABAP instances

This information describes the implementation of DB2 connection failover for ABAP type SAP instances.

The traditional SAP approach for implementing DB2 connection failover for ABAP instances was based on the SAP configuration file `connect.ini`. SAP calls this configuration option *SAP failover* (see “SAP failover for ABAP instances”). In earlier editions of this publication, this function was referred to as *SAP sysplex failover*.

The new SAP approach for implementing DB2 connection failover for ABAP instances is based on the native CLI driver failover capabilities that are delivered with DB2 Connect version 11.1 or later versions (see “CLI failover for ABAP instances”). SAP uses these capability through the configuration file `db2dsdriver.cfg`. SAP calls this configuration option *CLI failover*.

SAP Systems installed with Software Provisioning Manager (SWPM) 1.0 SP18 or newer versions do install CLI failover by default.

If your SAP systems are installed with SWPM 1.0 SP17 (or previous versions), then refer to the *Database Administration Guide: SAP on IBM DB2 for z/OS* for detailed instructions on how to convert existing installations from SAP failover to CLI failover.

The Software Update Manager (SUM) supports SAP installations that already use CLI failover.

SAP failover for ABAP instances

Previously, SAP employed a failover mechanism that was based on the SAP configuration file `connect.ini`. With DB2 12 and DB2 Connect version 11.1 this is no longer supported. The recommended failover mechanism is now **CLI failover** as described in the following. Details on the old mechanism can be found in *Business Continuity for SAP on IBM z Systems, Edition 12/2015 (SC33-8206-08)*.

CLI failover for ABAP instances

With **CLI failover**, SAP takes advantage of the native CLI driver failover capabilities that are delivered with DB2 Connect version 11.1 or later versions. **CLI failover** is installed by the SAP installer tool `SAPinst` for all new SAP installations on z/OS and it is a prerequisite for DB2 12.

From a technical point of view, the new and old failover mechanisms are managed in different software layers:

- The **CLI failover** is independent from SAP software. The failover is managed by the DB2 CLI driver and its failover feature. The failover configuration is defined in the `db2dsdriver.cfg` file.
- The **SAP failover** is managed by the SAP Database Interface and the SAP DBSL (Database Shared Library). The failover configuration like database locations, routing sequences and so on are defined in the so-called `connect.ini` file.

Figure 4 describes the differences between **SAP failover** and **CLI failover**.

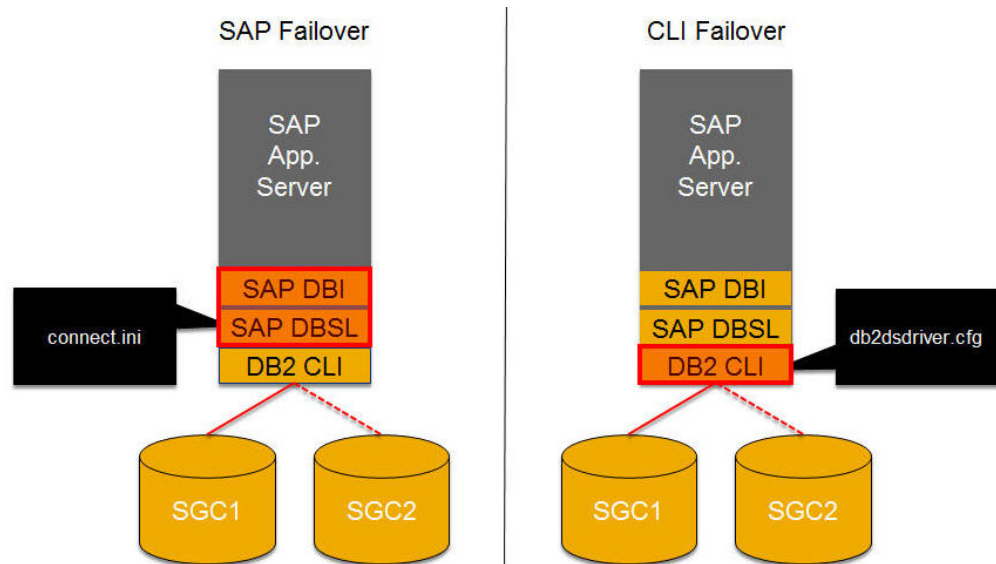


Figure 4. Differences between SAP failover and CLI failover

After the SAP installation with SWPM 1.0 SP 18 (or newer version), adaptations are required in the **CLI failover** configuration file `db2dsdriver.cfg` to achieve the desired failover mechanism. An example of such a **CLI failover** configuration file is explained in Chapter 12, “Operating an SAP system under System Automation control,” on page 243.

Changes in the CLI configuration file can be done either by an editor on operating system level, or by the SAP provided *Failover Configuration Tool* in SAP transaction DBACOCKPIT. For additional information refer to the SAP Community Network (SCN) article: *Setup of New DB2 Data-Sharing Failover Solution with the SAP Failover Configuration Tool*.

These are the main advantages of **CLI failover** in a planned failover scenario:

- A DB2 administrator can now initiate the process of a planned failover independently from an SAP administrator's support - using the same mechanisms for ABAP as for JAVA instances via STOP DDF.
- DB2 administrators can configure DB2 dynamic aliases and use the MODIFY DDF ALIAS(<alias-name>) STOP command to allow for a more granular control when initiating a planned failover. See also section *Setting Up the DB2 Distributed Data Facility (DDF)* in the *Database Administration Guide: SAP on IBM DB2 for z/OS*.

- Active DB2 threads reconnect to another DB2 member as customized in the configuration file `db2dsdriver.cfg`, which is an XML file in the SAP global directory.
- Long-running SAP batch jobs or dialog steps can seamlessly fail over to another DB2 member at a commit point even before the end of the job or dialog step is reached.
- From DBA Cockpit, you can dynamically reload changes in the **CLI failover** configuration file for all SAP application servers.
- With DB2 Connect 11.1, the **affinityFailbackInterval** parameter in `db2dsdriver.cfg` allows for a seamless failback of database connections to the primary DB2 member when this member is restarted after an planned or unplanned outage.

For unplanned failures, it is an important improvement that SAP transactions, which are in read-only state at the time the DB2 member fails, are seamlessly switched over from the failed DB2 member to another available member. So these SAP transactions do not receive a negative SQL code and can continue processing.

CLI failover provides an hourly DB2 thread recycle on the DB2 server side. (For details see the description in APAR PK69339). The thread recycling is triggered in order to free up resources on the DB2 server side. The thread recycling is transparent to SAP. It has no impact on SAP work processes or on SAP transactions.

The following warning message in the SAP developer work files `dev_w<x>` indicates that a thread recycling has occurred:

```
DB2 Call 'SQLExecute'
Warning: SQLCODE = 89999 : [IBM][CLI Driver]
CLI0212W
A seamless failover occurred during a connect or an execute request.
SQLSTATE=01000
```

With **SAP failover**, the recommendation was to trigger an SAP work process reconnect every 24 hours by setting the profile parameter `rdisp/wp_auto_restart` to 86400 seconds. This is no longer necessary when using **CLI failover**.

DB2 connection failover for Java instances

This information describes the implementation of DB2 connection failover for Java type SAP instances.

DB2 connection failover for SAP Java instances is based on the native database failover feature of the *IBM Data Server Driver for JDBC*, which is packaged as part of DB2 Connect.

DB2 Connect 9.7 Fix pack 1 introduced a configuration capability with the help of an XML connection profile, and also introduced several further features such as *seamless failback*. SAP Java instances exploit this capability, see *Database Administration Guide: SAP on IBM DB2 for z/OS*. This document also describes how to install the JDBC driver and how to configure the XML connection profile for the following purposes:

Seamless failover in case of planned outages

If a DB2 or system maintenance on one LPAR is required, you can move the Java workload off a specific DB2 member: Stop the DDF on this DB2 member by issuing `STOP DDF MODE(QUIESCE)`.

At the end of an SQL transaction, DB2 threads are redirected to the next DB2 member as configured in the configuration XML. If no resources are held anymore in the DB2 member, you can stop the DB2 member completely and perform your maintenance task.

Configuration for unplanned outages

After an unplanned DB2 or system outage, you use the configuration XML to define to which DB2 members the failover and reconnect of the SAP Java workload occurs.

Seamless failback

When a DB2 data sharing member becomes available again after a planned or unplanned outage, the Java workload that was redirected to another DB2 member, returns to its original DB2 member to achieve optimal workload distribution.

SAP Java workload distribution

If you have more than one Java application server, you might want to distribute their workload to different DB2 members. This workload distribution can also be accomplished with the help of the configuration XML.

Building blocks for DB2 connection failover

The contained information describes the basic building blocks that are needed for DB2 connection failover. It introduces the two major data sharing failover configurations for providing a highly available environment for an SAP database on DB2 for z/OS.

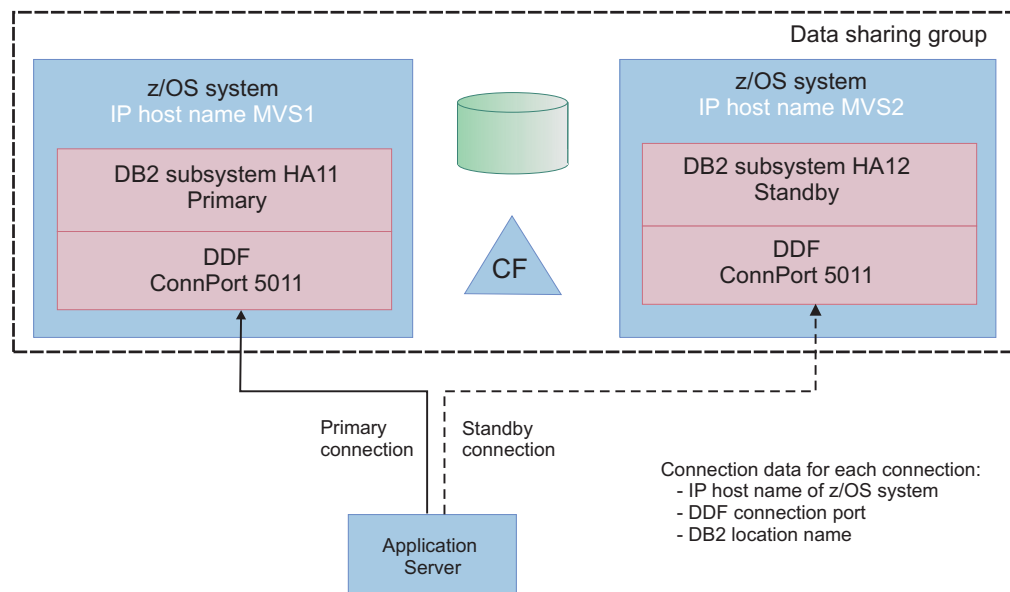


Figure 5. SAP DB2 connection failover configuration: Active/Passive example

Two major data sharing failover configurations provide a highly available environment for an SAP database on DB2 for z/OS:

- Active/Passive: Single DB2 member with passive (inactive) standby member
- Active/Active: Two or more active DB2 members in active-standby mode

To enable a reuse of system resources during maintenance, both configurations can be enhanced with a passive standby member for each active member to allow for cascaded failover. Cascaded failover means that first the passive member is used for failover, which resides on the same CEC (primary passive). In case that the CEC is down, for maintenance for example, the failover is done to either the secondary passive or the secondary active.

For more information about data sharing failover configurations, for example, cascaded failover to reuse the MIPS on the primary CEC, see *Database Administration Guide: SAP on IBM DB2 for z/OS* .

The sample SAP system name (SID) HA1 is used in Figure 5 on page 19. In the example, HA1 is used as the DB2 group name.

The sample introduces the notion of *primary DB2 members* and *standby DB2 members*. *Primary DB2 members* normally have application servers attached that are doing productive work. *Standby DB2 members* can run in hot standby mode with no attached application servers.

- The primary DB2 member names are the DB2 group name plus a digit (for example, **HA11**).
- The standby DB2 member names consist of the DB2 group name plus a digit (**HA12**).

Figure 5 on page 19 illustrates an implementation in which each application server has a primary DB2 member in one LPAR (MVS1) and a standby DB2 member in a standby LPAR (MVS2).

To connect the ABAP or Java application servers to the standby DB2 member, the described DB2 Connect failover mechanisms use:

- the DB2 location name,
- the z/OS LPAR virtual IP host name,
- the connection port.

In the event of a planned or unplanned incident, the SAP Database Shared Library (DBSL) recognizes the need to fail over. In the case of SAP failover, it then looks for standby information in the `connect.ini` control file, and attempts to connect the application server to a standby DB2 member. In the case of CLI failover, the CLI driver itself uses the information specified in the `db2dsdriver.cfg` file to connect to a standby DB2 member.

Chapter 3. Planning for high availability for SAP

The solutions, which are described in this publication use the autonomic computing technologies of IBM z Systems products to provide automation and high availability for SAP systems. This information is intended to give an overview of what must be considered when you want to make an SAP system highly available.

High availability means the ability to avoid planned and unplanned outages by eliminating single points of failure. This ability 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 hiding the outage from the users. This ability can be accomplished by quickly restarting failed components with the functions of the *System Automation product family*. System Automation uses sophisticated, policy-based knowledge about the SAP application components and their relationships. Based on an availability goal for the SAP application, this knowledge is used to

- automatically start or stop a complete SAP system.
- decide on corrective actions if a planned or unplanned outage takes place.

It is highly recommend using Tivoli System Automation to implement high availability for SAP on z Systems. However, you can use optional intermediate stages of availability as described in “Possible preparation stages of high availability for SAP.” Each subsequent stage builds on the predecessor stage. With “Stage 3: Maximum availability” on page 23, you finally achieve a full-blown high availability implementation. To implement stage 3, you can select from options 1 and 2, as described in “How to decide which System Automation option to implement” on page 24. Note that for Windows application servers only *Option 1* is selectable.

If you want to implement “Option 2: Automation by System Automation products with central components on AIX or Linux” on page 26, you need to implement only parts of the intermediate preparation stages 0 and 1 (base and limited availability, described in “Possible preparation stages of high availability for SAP”). The reason is, that you have no SAP Central Services running under z/OS.

Possible preparation stages of high availability for SAP

Read a description of the preparation stages that you can implement before you select one of the high availability options that are described in this publication. Implementing these stages reduces the effort to establish a full-blown high availability implementation.

Stage 0: Base availability

Through implementing this stage, the most important single points of failure (SPoF) of the SAP architecture, namely the database server, and the NFS server (global SAP file systems) are eliminated.

- Set up at least two DB2 data sharing members. For DB2 automation, it is highly recommend to use Tivoli System Automation for z/OS, which comes with a best practice policy for DB2.
- Set up an NFS server on z/OS, which requires automation for failover and high availability. Tivoli System Automation also offers best practice policy for the NFS server:
 - Use an own VIPA for the NFS server.
 - Use a zFS shared across the sysplex.
- Install SAP Java and ABAP Central Services instances (Java SCS and ASCS) on z/OS with ZSCSinst:
 - Use Virtual IP addresses (VIPAs) for each installed instance.
 - Do not install any ERS instance.

Note: You must define CF structures in order to enable EnqCF replication. See *SAP Note 1753638: z/OS: Enqueue Replication into System z Coupling Facility* and the attached PDF file for details.

Stage 1: Limited availability

Through implementing this stage, the important SPoF of the SAP architecture, the SAP Central Services, is partially eliminated.

Use the moddvipa command to start the SCS-specific VIPA before you start the instance itself.

Use the SAP build-in automation that is based on the most current kernel.

- Adapt the SCS profile according to *SAP note 768727: Process automatic restart functions in sapstart*: Change Start_Program_xx to Restart_Program_xx. This profile option tells SAP to restart the enqueue, message, and gateway servers in place.
- Use the SAP startsapsrv framework to start and stop the SCS instances:
 1. Start the sapstartsrv service.
 2. Start the instance itself to instantiate the following process hierarchy:
 - sapstart:
 - enqueue server
 - message server
 - gateway server

Availability benefits compared to stage 0:

- If the enqueue, message and gateway servers fail, they are restarted by SAP's sapstart process.

Disadvantages:

- The contents of the enqueue table is lost.
- SAP transactions are rolled back.
- The user must reenter data into the SAP GUI.
- Batch job activities must be manually reset and rerun.
- Only a manual failover is possible, but no automation.

Stage 2: Medium availability

Through implementing this stage, the important SPoF of the SAP architecture, the SAP Central Services, is eliminated for planned maintenance activities.

Use moddvipa command to start the SCS-specific VIPA before you start the instance itself.

Use the SAP build-in automation that is contained in SAP Kernels 7.2x and higher, and also use replication into the coupling facility.

- Adapt the SCS profile according to *SAP note 768727: Process automatic restart functions in sapstart*: Change Start_Program_xx to Restart_Program_xx. This profile option tells SAP to restart the enqueue, message, and gateway servers in place.
- Additionally, activate and use the enqueue replication into coupling facility. This functionality requires SAP kernel 7.21 or higher.
- Use the SAP startsapsrv framework to start and stop the SCS instances:
 1. Start the sapstartsrv service.
 2. Start the instance itself to instantiate the following process hierarchy:
 - sapstart:
 - enqueue server
 - message server
 - gateway server

Availability benefits compared to stage 1:

- If the enqueue, message or gateway servers fail, they are restarted by SAP's sapstart process.
- If the enqueue server can be restarted in-place:
 - The contents of the enqueue table is not lost and is rebuilt.
 - No transactions are rolled back.
 - The restart is nearly imperceptible for the SAP user and is therefore non-disruptive.

Disadvantages:

- There is no automatic failover, for example, in case of an unplanned outage, or if the restart in place of SAP failed.

Stage 3: Maximum availability

If you implemented stages 1 and 2, you can now implement “Option 1: Automation using System Automation for z/OS only” on page 24 to obtain full-blown automation.

If you want to implement “Option 2: Automation by System Automation products with central components on AIX or Linux” on page 26, then you do not have SCS running under z/OS. But you can implement parts of the intermediate stages 0 and 1 (base and limited availability).

Through implementing any of these options, all SPoFs of the SAP architecture are eliminated for planned and unplanned outages.

Availability benefits compared to stage 2:

- If the enqueue, message or gateway servers fail, they are restarted by SAP.
- The contents of the enqueue table is not lost and is rebuilt.
- No transactions are rolled back.
- There is an indiscernible restart in place of the enqueue server, if restart is possible.

- If a restart in place fails, System Automation initiates an automatic failover to another LPAR. This process is a two-level automation: No transactions are rolled back, and there is an invisible failover plus an automatic failover in case of planned and unplanned outages of an LPAR, CEC, and so on.

How to decide which System Automation option to implement

Two different configuration options (also referred to as HA Option 1 and 2, or just Option 1 or Option 2) are described that offer different levels of automation and availability. During the planning phase of your high availability solution, you select and decide which option you want to implement. The decision depends mainly on your business continuity requirements.

- “Option 1: Automation using System Automation for z/OS only” places all vital components on the z/OS operating system. Also, this option uses System Automation for z/OS as the focal point to manage both the local SAP resources on z/OS and the remote application server resources. It integrates with SAP's `sapstartsrv` framework and infrastructure.

Note:

- HA Option 1 is the recommended configuration implementation that concentrates on SA z/OS.
- For Windows application servers only this option is selectable.
- “Option 2: Automation by System Automation products with central components on AIX or Linux” on page 26 offers a similar level of automation as Option 1, but places all vital components except the database on AIX or Linux operating systems. For SAP Central Services, it does integrate with SAP's `sapstartsrv` framework and infrastructure. This option requires the use of SA z/OS for automating z/OS resources and System Automation for Multiplatforms to automate AIX or Linux resources.

Important: Alternatively you can still use a mix of SAP resources controlled by SA z/OS and SA MP. For example, you can manage SAP Central Services under SA z/OS and SAP Application server under SA MP. Such a setup is no longer described as an own HA option.

Note: In the following, the phrase *integrates with SAP's `sapstartsrv` framework and infrastructure* also implies that the **SAPSRV add-on policy* enables seamless integration of an SA z/OS controlled SAP Central Services instance into SAP systems management tools, like SAP's Management Console, or into SAP life-cycle tools, like the SAP Software Update Manager (SUM).

Option 1: Automation using System Automation for z/OS only

HA Option 1 places all vital components on the z/OS platform. Also, this option uses System Automation for z/OS as the focal point to manage the local SAP resources on z/OS and the remote application server resources. It integrates with SAP's `sapstartsrv` framework and infrastructure. The environment is automated with System Automation only. Note that this is the only supported option, if you want to include a Windows application server under System Automation control.

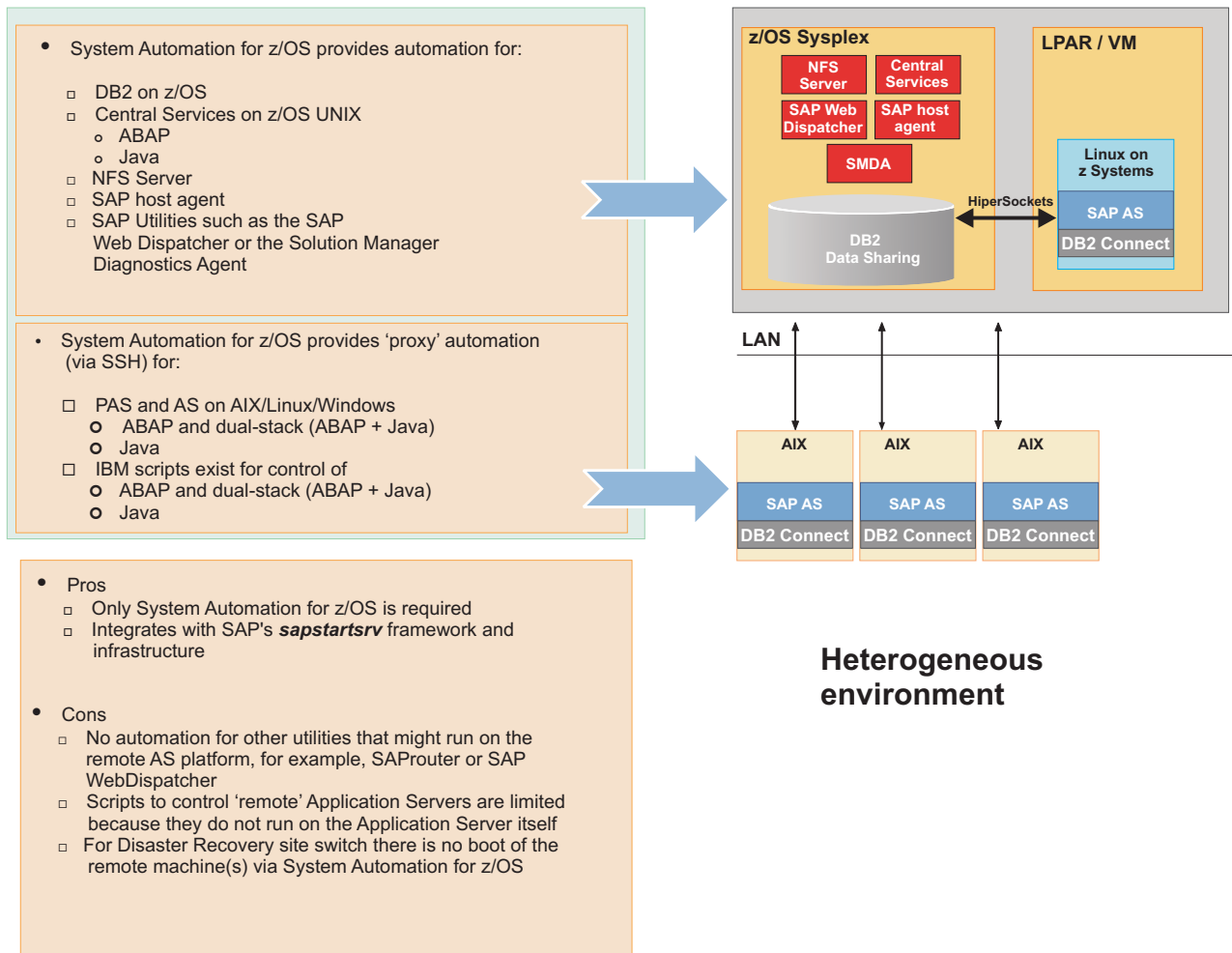


Figure 6. Option 1: Automation using SA z/OS only

The ABAP or Java application servers are controlled via proxy resources.

Option 1 uses SA z/OS to control and automate all SAP-related components that run under z/OS or z/OS UNIX. In addition, SA z/OS also controls the remote SAP application server instances through proxy resources. The definition of these proxy resources must have the following features:

- Include mechanisms to issue SSH commands to the remote machines to start and stop the application servers. SSH has the advantage that it can be set up using keys for authorization, which enables transparent log-in on a remote application server. Transparent means, that a script can log in without providing a password or a password phrase.
- Include mechanism to use TCP/IP - based monitoring via the `samsapctrl_asping` utility (see "ABAP application server instances" on page 137 and "Java and dual-stack application server instances" on page 141).

Therefore, SA z/OS controls:

- DB2 z/OS data sharing members
- SAP Central Services (for ABAP, or for Java, or for both)
- Application servers on AIX, Linux, or Windows via proxy resources

- NFS server
- SAP host agent
- Other SAP utilities that can run under z/OS or z/OS UNIX, for example, SAP Web Dispatcher, or the Solution Manager Diagnostics Agent.

Non-z/OS SAP resources like other utilities that have an affinity to the application server are outside the control of SA z/OS. These resources are not highly available.

The sample scripts that are used to control the application servers on AIX, Linux, or Windows do not run on the application server itself and need SSH, for example, to execute commands on the application servers.

This option requires that you set up and configure SA z/OS.

This configuration implies that you have SAP components that are running on two different platforms:

- the SAP Central Services on z/OS UNIX, and
- the SAP application servers on AIX, Linux, or Windows.

Option 2: Automation by System Automation products with central components on AIX or Linux

HA Option 2 places all vital components except the database on AIX or Linux operating systems. For SAP Central Services, it integrates with SAP's sapstartsrv framework and infrastructure. The complete SAP environment is automated with the help of Tivoli System Automation for z/OS and Tivoli System Automation for Multiplatforms.

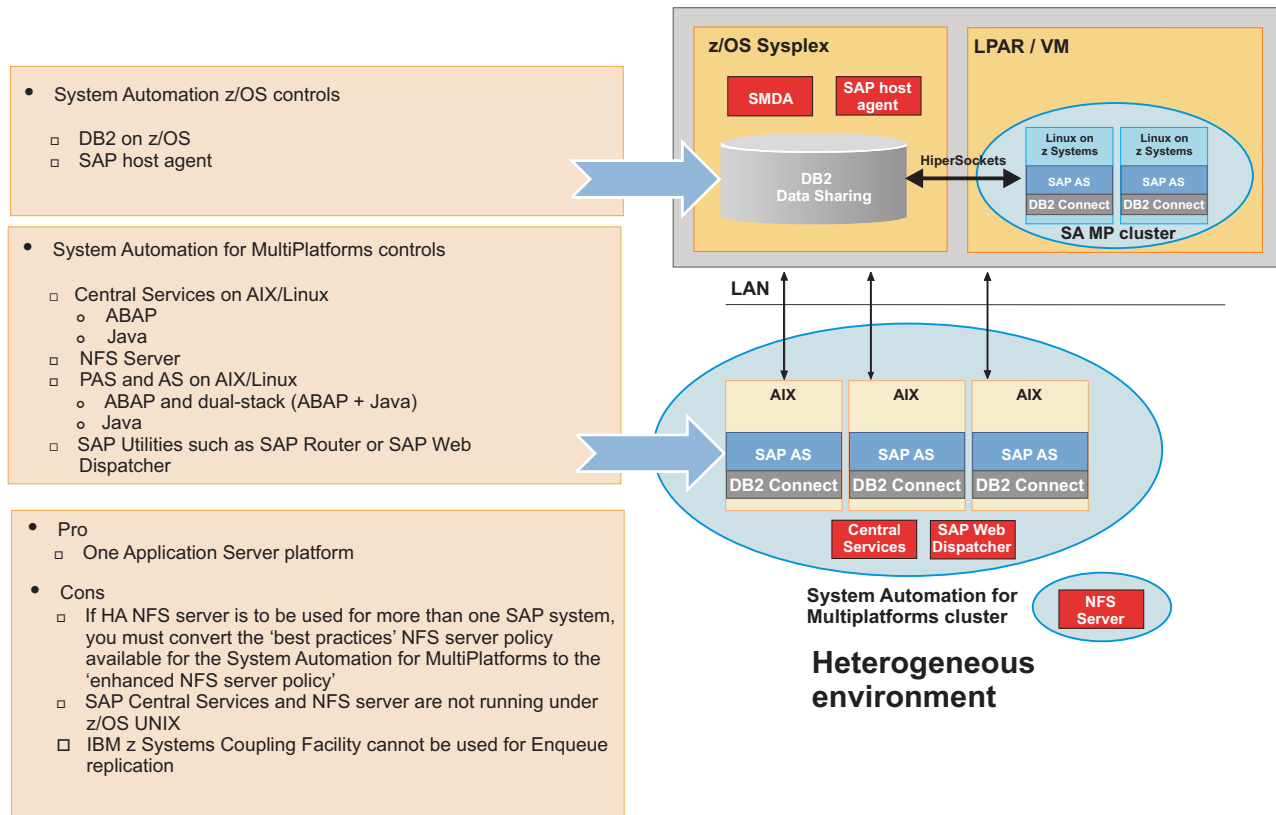


Figure 7. Option 2: Automation with the support of SA z/OS and Tivoli System Automation for Multiplatforms

Option 2 differs from Option 1 in that the SAP Central Services and NFS server are running under the control of System Automation for Multiplatforms.

In Option 2, SA z/OS manages:

- DB2 z/OS data sharing members
- SAP host agent
- Other SAP utilities capable of running under z/OS or UNIX System Services.

In Option 2, System Automation for Multiplatforms manages the following SAP components:

- SAP Central Services (for ABAP or for Java, or both)
- The primary and additional SAP application servers (PAS and AS)
- NFS server
- SAP utilities, like for example, the SAP Web Dispatcher

In addition, other utilities (with an affinity to application servers) are automated in a cluster that is managed by Tivoli System Automation for Multiplatforms.

For information about how to set up the highly available NFS server, see Step “5. Making NFS highly available” on page 32.

To set up Option 2, you must set up and configure:

- Tivoli System Automation for z/OS
- Tivoli System Automation for Multiplatforms

This configuration implies that all SAP components (SAP Central Services and the SAP application servers) run on one common operating system: AIX or Linux.

Technologies on IBM z Systems for highly available SAP solutions

To apply availability technologies, you need to complete certain planning and implementation steps for your solution.

1. Use DB2 on IBM z Systems in data sharing mode as the database server.
2. Configure the network between application servers and database server to be highly available. This configuration can be either:
 - Hardware-based, if paired switch equipment is used, for example, such as Cisco's Virtual Switching System
 - Software-based, by using features like z/OS virtual IP address (VIPA) and dynamic routing (OSPF) to get a fault-tolerant, highly available network
3. Take advantage of SAP features that support high availability.
 - SAP Central Services with replication
 - DB2 connection failover
 - SAP logon groups
4. Automate system operations for SAP, which means automate the start, stop, restart, and failover of SAP components and all required z/OS components.
5. Ensure to have a highly available NFS server.

1. Use DB2 data sharing on z Systems as database server

To achieve high availability, it is important to run DB2 database server in data sharing mode. DB2 in data sharing mode is a true parallel database server, and ensures redundancy of the database at the highest level. Additionally, online backup and online recovery features, which are coupled with the features of the z Systems platform, provide high availability and also provide DB2 rolling maintenance and release upgrade.

Two major data sharing Parallel Sysplex failover configurations on z Systems provide a highly available environment for an SAP database on DB2 for z/OS:

- Active/Passive: Single active DB2 member with passive standby member
- Active/Active: Two or more active DB2 members

For more information about each of these options, see “DB2 data sharing on z Systems Parallel Sysplex” on page 13.

If you use, or plan to use, IBM DB2 Analytics Accelerator, and want to integrate the accelerator into existing high availability architectures and disaster recovery processes, refer to the following redbook:

IBM DB2 Analytics Accelerator: High Availability and Disaster Recovery

2. Configure the network

In a highly available network, all network components of the physical layer (network adapters and network control equipment, for example, switches, and cables) must be eliminated as a single point of failure. This elimination can be achieved by duplicating all network components to create redundancy. Thus you have at least two different and independent physical network paths to the z/OS database server from each remote SAP application server.

In failure situations, the duplicate paths must be used in a non-apparent way from an application point of view. The appropriate method can be either hardware-based, if paired switch equipment is used, or software-based, if features, such as z/OS virtual IP address (VIPA) and dynamic routing (OSPF) are used. Such a software-based setup is mandatory if you want to exploit HiperSockets™ because HiperSockets cannot be used for communication between different physical systems (CECs).

Software-based network high availability builds on implementing:

- Open Shortest Path First (OSPF) protocol as a recovery mechanism
OSPF is a dynamic link-state routing protocol. It aids recovery of TCP/IP connections from network failures by finding an alternative path to the destination. The IP layer then uses this path to route IP packets to the destination. Compared to other routing protocols, OSPF updates its routing table faster and has a shorter convergence time.
- Virtual IP address (VIPA) as a recovery mechanism
A VIPA is an IP address that is associated with a TCP/IP stack and is not tied to a physical interface. Therefore, it is less likely to fail. It can be reached through any of the physical interfaces of that TCP/IP stack and it is advertised to the IP routers by dynamic routing. Therefore, if one of the physical network interfaces fails, the VIPA can still be reached through one of the other network interfaces. Thus, this single point of failure is eliminated.

Hardware-based network high availability is based on (proprietary) mechanisms of the switch providers that handle switch failures.

Chapter 4, “Network characteristics for high availability,” on page 39 describes these options in detail.

An example of a recommended setup can be found in “Recommended setup for high availability connections between client and server” on page 48

3. Use SAP features supporting high availability

DB2 connection failover provides the capability to redirect application servers to a standby database server, if the primary database server becomes inaccessible. By using the DB2 data sharing function in a sysplex, you can provide redundancy at the database service layer. For more information, see “DB2 data sharing on z Systems Parallel Sysplex” on page 13.

Install and run the stand-alone enqueue and enqueue replication server for ABAP and Java. For this purpose, use SAP Central Services for ABAP and Java (ASCS and Java SCS). For details, see Chapter 7, “Customizing SAP for high availability,” on page 113.

Install each SAP Central Service instance with its own virtual host name so that each instance can be separately moved within a cluster, if necessary. Stand-alone enqueue servers avoid enqueue data loss and database rollbacks during a failure of the central enqueue server. It allows fast failover of those services to a standby machine where the replication server runs. Using the *EnqCF replication* mechanism (see Chapter 13, “Enqueue replication into a z Systems coupling facility,” on page 277), your SAP Central Services no longer require a separate replication server. In addition, an SAP Central Services installation is a prerequisite for a rolling kernel update, which helps you to further reduce planned outages (see *SAP Note 953653: Rolling kernel switch*).

Another SAP function to exploit is the use of *SAP logon groups*. This enables the distribution of SAP users across available application servers based on requirements for workgroup service and utilization. In case an application server becomes unavailable. An SAP user drops out of his SAPGUI session and must relogin in order to be transferred to another application server.

In addition to ensuring that an SAP application server (ABAP or JAVA) instance is started, you can check its availability:

- For the ABAP and Java application servers, use the `samsapctrl_asping` utility, which uses the `sapcontrol` interface to get the application server health status. For a detailed description of `samsapctrl_asping`, see “ABAP application server instances” on page 137.

4. Automate system operations for SAP

IBM Tivoli System Automation for z/OS (SA z/OS) as well as Tivoli System Automation for Multiplatforms belong to a product family that provides high availability by automating the control of resources such as processes, file systems, and IP addresses in z/OS, Linux or AIX-based clusters. It facilitates the automatic switching of applications and data from one system to another in the cluster after a hardware or software failure.

To cope with the operational complexity of an SAP environment and to avoid operator errors, it is advisable to automate system operations for SAP with the System Automation product family. Automation also keeps the SAP system up and running when failed resources are detected, for example, through restart or failover. Bearing that in mind, automation also means automating the start, stop, restart, and failover of SAP components to give the SAP environment the wanted high availability.

The following is a list of such resources and components. It is not a complete list, but instead gives an overview of what you need to consider. However, you are not required to define these resources and components *from scratch*. The system automation products come with so-called *best practice* policy samples, which you can adapt to your real environment.

Operating system components

- z/OS, Linux, or AIX
- Network components, including
 - TCP/IP (Linux: part of kernel)
 - VIPA/Source VIPA
 - OSPF routing
 - NFS
- File system

Database

- DB2

Parallel Sysplex

- Integration with GDPS

SAP components

- ABAP Central services:
 - Enqueue server
 - Message server
- Java central services:

- Enqueue server
- Message server
- Java Gateway (required since SAP 7.1)
- ABAP enqueue replication server (if applicable)
- Java enqueue replication server (if applicable)
- Application server
 - Remote ABAP or Java instances
- Other components:
 - SAP host agent
 - SAProuter
 - SAP Web Dispatcher

You must perform the following SAP customization steps during and after installation if you plan to run SAP under the control of system automation:

1. Install SAP in a high availability setup. For example, install all SAP Central Services (SCS) with their *own virtual hostname*. Even for an SAP dual-stack installation with an ABAP and a Java SCS instance, each instance must be installed with its own virtual host name.
2. Add SAP components that are required to achieve high availability of your SAP system but were not installed by the SAP installer. For example, earlier versions of the SAP 7.0 installer might not install the SAP enqueue replication server.

The installation and setup of the previous mentioned components is mainly a manual task. For system automation, *best practice* policies are available for modeling an SAP system. Use these *best practice* policies as a starting point to create your own system automation policies. With SA z/OS, use the **SAPSRV add-on policy* and the SAP High Availability wizard (SAP HA wizard) to automate SA z/OS policy generation. Because each SAP environment is unique, the output of the SAP HA wizard requires manual post-processing and subsequent testing.

Refer to the *developerWorks: SAP on IBM z Systems Community* for samples that show how to invoke the SAP HA wizard.

At a high level, the following steps are required to implement SAP automation and high availability with the System Automation family:

1. Install and set up SA z/OS and Tivoli System Automation for Multiplatforms, and define all z/OS base subsystems to SA z/OS. See Chapter 6, “Preparing a high availability SAP solution,” on page 89.
2. Make the NFS server highly available via system automation:
 - For details about how to set up a high availability NFS server under z/OS, see Chapter 6, “Preparing a high availability SAP solution,” on page 89.
 - For details of how to set up a high availability NFS server under AIX or Linux, see Chapter 9, “Customizing System Automation for Multiplatforms,” on page 185.
3. Install and customize SAP with a high availability setup with the help of virtual host names for each SAP Central Service instance. See Chapter 7, “Customizing SAP for high availability,” on page 113.
4. Make sure you can start and stop all SAP components manually on all systems on which the components are intended to run. Verify that the SAP components operate correctly.
5. Move SAP components under system automation control under z/OS as well as under Tivoli System Automation for Multiplatforms according to the *best*

practice SAP policy for your *selected automation option* as described in “How to decide which System Automation option to implement” on page 24.

6. Set up Tivoli System Automation. For all system automation-related tasks, refer to topics Chapter 8, “Customizing Tivoli System Automation for z/OS,” on page 145 and Chapter 9, “Customizing System Automation for Multiplatforms,” on page 185.
7. Verify that the installation works as expected. See topic Chapter 11, “Verifying your System Automation for Multiplatforms implementation on Linux or AIX,” on page 241.

5. Making NFS highly available

NFS is a part of a distributed SAP system and as such must also be highly available. For the highest level of availability, it is recommended to run the NFS server on z/OS under SA z/OS control.

Alternatively, it is also possible to run the NFS server outside z/OS (see “Option 2: Automation by System Automation products with central components on AIX or Linux” on page 26). You can set up a highly available NFS server under Tivoli System Automation for Multiplatforms in either of the following ways:

- Run the NFS server within the *same* Tivoli System Automation for Multiplatforms domain as the SAP system. Use this option if you want to use the NFS server just for that single SAP system.
- For the NFS server define its own Tivoli System Automation for Multiplatforms cluster. Use this option if the NFS server is to serve several SAP Systems. In this case, use the *NFS high availability policy* of Tivoli System Automation for Multiplatforms.

Disaster recovery

Read the contained information about disaster recovery aspects that are related to SAP on DB2 for z/OS.

For more information about the contained topics, see the IBM documentation and redbooks, like for example, *IBM GDPS Family: An Introduction to Concepts and Capabilities* (<http://www.redbooks.ibm.com/abstracts/sg246374.html>).

A comprehensive description of the SAP backup and recovery procedures, both for data sharing and non-data sharing, is given in the SAP Community Network (SCN) document *Casebook 2014 Edition: Tightly Integrated DB2 Backup, Recovery and Cloning for SAP Environments*, which is available at: <http://scn.sap.com/docs/DOC-24881>.

Data sharing considerations for disaster recovery

If you introduce a disaster recovery solution, it is usually based on a DB2 data sharing implementation. The provided information describes the most important concepts and different options for implementing a disaster recovery strategy with data sharing, ranging from the traditional method to the most up-to-date implementation.

The options for implementing a disaster recovery strategy with data sharing are essentially the same as the options in non-data sharing environments. However, some new steps and requirements must be addressed.

Specific information about data sharing is available in *DB2 for z/OS Data Sharing: Planning and Administration* (SC19-2973).

Configuring the recovery site

If the distance between the primary site and the secondary site is too far to run a stretched DB2 data sharing group with members on both sites, the recovery site must have a data sharing group that is identical to the group at the local site. It must have the same name and the same number of members, and the names of the members must be the same. The coupling facilities resource manager (CFRM) policies at the recovery site must define the coupling facility structures with the same names, although the sizes can be different. You can run the data sharing group on as few or as many z/OS LPARs as you want.

If you have configured SAP enqueue replication into the z/OS coupling facility (see Chapter 13, “Enqueue replication into a z Systems coupling facility,” on page 277), then you need to make sure that the *coupling facilities resource manager* (CFRM) policies at the recovery site include the CF structures that are needed for this mechanism.

The hardware configuration can be different at the recovery site as long as it supports data sharing. Conceptually, there are two ways of running the data sharing group at the recovery site. Each way has different advantages that can influence your choice:

- Run a multisystem data sharing group.
The local site is most likely configured this way, with a Parallel Sysplex, which contains many CECs, z/OS LPARs, and DB2 subsystems. This configuration requires a coupling facility, the requisite coupling facility channels, and the Server Time Protocol (STP).

The advantage of this method is that it has the same availability and growth options as on the local site. In general, it is recommended to use this method to keep the high availability characteristics of your SAP solution when it is running on the secondary site.

- Run a single-system data sharing group.
In this configuration, all DB2 processing is centralized within a single z Systems server, which can support the expected workload. Even with a single CEC, a multi-member data sharing group, which is using an internal coupling facility must be installed. After the DB2 group restart, all but one of the DB2 members are shut down, and data is accessed through that single DB2.

Obviously, this approach loses the availability benefits of the Parallel Sysplex, but the single-system data sharing group has fewer hardware requirements:

- An STP is not required as the CEC time-of-day clock can be used.
- Any available coupling facility configuration can be used for the recovery site system, including Integrated Coupling Facilities (ICFs).

With a single-system data sharing group, there is no longer R/W interest between DB2 members, and the requirements for the coupling facility are:

- a LOCK structure (which can be smaller)
- an SCA

Group buffer pools are not needed to run a single-system data sharing group. However, small group buffer pools are needed for the initial start-up of the group

so that DB2 can allocate them and perform damage-assessment processing. When it is time to do single-system data sharing, remove the group buffer pools by stopping all members. Then restart the member that is handling the workload at the disaster recovery site.

GDPS infrastructure for disaster recovery

GDPS is an abbreviation for Geographically Dispersed Parallel Sysplex. It is a multi-site application that provides the capability to manage:

- the remote copy configuration and storage subsystems
- automated Parallel Sysplex tasks
- failure recovery

Its main function is to provide an automated recovery for planned and unplanned site outages. GDPS maintains a multisite sysplex, in which some of the z/OS LPARs can be separated by a limited distance. GDPS adheres to the sysplex specification in that it is an application-independent solution.

The primary site contains some of the z/OS LPARs supporting some of the data sharing group members, and the primary set of disks. These disks are the ones that support all DB2 activity that comes from any DB2 member of the group. At the secondary site, there are active sysplex images, which support active DB2 members that are working with the primary set of disks. There is also a secondary set of disks, which are mirror copies of the first site.

GDPS supports three data mirroring technologies:

1. Metro Mirror (formerly Peer-to-Peer Remote Copy (PPRC)) in which
 - the mirroring is synchronous
 - GDPS manages secondary data consistency and therefore no, or limited, data is lost in failover
 - the production site performs exception condition monitoring. GDPS initiates and processes failover
 - distances between sites are up to 40 km (fiber)
 - both are provided: continuous availability and disaster recovery solution
2. z/OS Global Mirror (formerly XRC) with:
 - asynchronous data mirroring
 - limited data loss is to be expected in unplanned failover
 - Global Mirror managing secondary data consistency
 - GDPS running Parallel Sysplex restart
 - supporting any distance
 - providing only a disaster recovery solution
3. Global Mirror with:
 - asynchronous data mirroring
 - disk-based technology
 - supporting any distance
 - supporting a mix of CKD and FBA data

In addition to GDPS, there is also an entry-level offering that consists of Tivoli Storage Productivity Center for Replication (TPC-R) exploiting z/OS Basic HyperSwap. Basis HyperSwap masks primary disk storage system failures by transparently switching to the secondary disk storage system. This means it is

non-disruptive and is designed for thousands of z/OS volumes. However, it is not intended for advanced disaster recovery scenarios, for example, no data consistency for cascading primary volume failures. For this purpose, use GDPS.

The following is an example of multifunctional disaster recovery infrastructure which uses GDPS and Metro Mirror to provide all the elements of a backup and recovery architecture. It includes

- conventional recovery, to current and to a previous point in time
- disaster recovery
- fast system copy capability to clone systems for testing or reporting
- forensic analysis system (a corrective system as a "toolbox" in case of application disaster)
- compliance with the high availability requirements of a true 24x7 transaction environment that is based on SAP

This configuration is prepared to support stringent high availability requirements in which no quiesce points are needed.

The non-disruptive DB2 *BACKUP SYSTEM* utility is used to obtain backups without production disruption. No loss of transactions and data is encountered. The infrastructure provides for a forensic analysis system as a snapshot of production that can be obtained repeatedly throughout the day.

The components of this sample solution are IBM z Systems, z/OS Parallel Sysplex, DB2 for z/OS data sharing, GDPS with automation support, IBM DS8000® disk subsystems with Metro Mirror/Global Mirror and FlashCopy® functions, and SAP/IBM replication server for high availability of the applications.

The following figure shows the GDPS solution landscape.

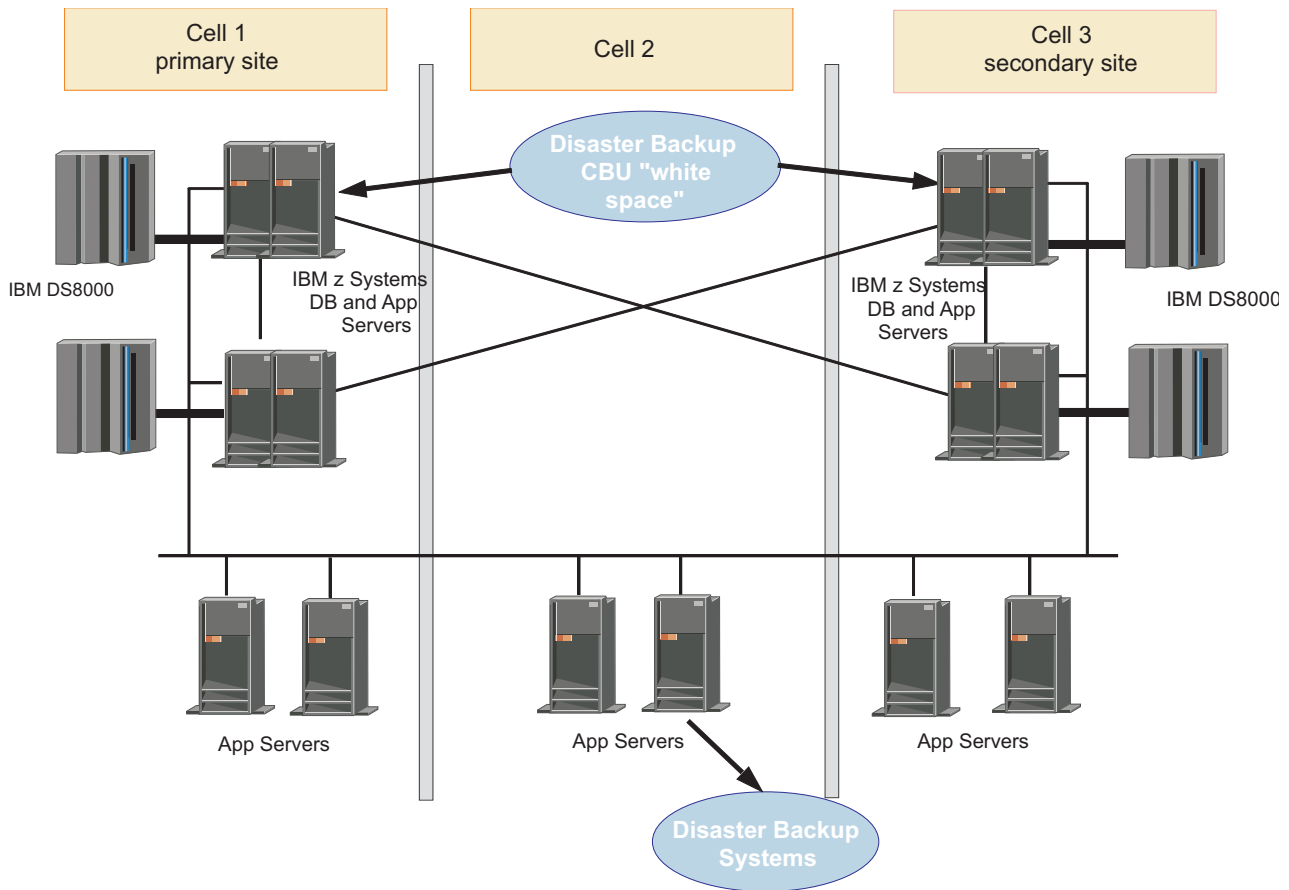


Figure 8. Example of high availability with GDPS configuration

This configuration is made up of two sites and three cells. (Cell 2 is where the corrective system is started.) The three cells are encapsulated and safe against floods and earthquakes, for example. The distance between cell 1 and cell 3 should be about 20 km based on GDPS recommendations. Both cells belong to the same sysplex and keep members of the same data sharing group. Cell 2, on the other hand, is out of the sysplex to keep the same DB2 data set names for the corrective system.

DS8000 primary and active set of disks is located on the primary site and, using Metro Mirror, the disks are mirrored to the secondary site. Because the BACKUP SYSTEM utility is used, it is not necessary to split the mirror to get a non-disruptive backup. The design keeps symmetry between both sites, with the same DS8000 disk capacity on each site. Therefore, if one site is not available (disaster, maintenance), the other is able to provide an alternate backup process.

Remote site recovery using archive logs

If you are not using GDPS, you can consider using the following approach. Apart from these configuration issues, the disaster recovery procedural considerations do not affect the procedures already put in place for a single DB2 when enabling data sharing. All steps are documented in the *DB2 for z/OS Administration Guide* (SC19-4050).

The procedure for DB2 data sharing group restart at the recovery site differs in that there are steps, which ensure that group restart takes place in order to rebuild

the coupling facility structures. In addition, you must prepare each member for conditional restart rather than just a single system.

To force a DB2 group restart, you must ensure that all of the coupling facility structures for this group have been deallocated:

1. Enter the following MVS™ command to display the structures for this data sharing group:

```
D XCF,STRUCTURE,STRNAME=grpname*
```

2. For the LOCK structure and any failed-persistent group buffer pools, enter the following command to force the connections off of those structures:

```
SETXCF FORCE,CONNECTION,STRNAME=strname,CONNNAME=ALL
```

With group buffer pools, after the failed-persistent connection has been forced, the group buffer pool is deallocated automatically.

To deallocate the LOCK structure and the shared communication area (SCA), it is necessary to force the structures out.

3. Delete all of the DB2 coupling facility structures by using the following command for each structure:

```
SETXCF FORCE,STRUCTURE,STRNAME=strname
```

This step is necessary to clean out old information that exists in the coupling facility from your practice startup when you installed the group.

The following is a conceptual description of data sharing disaster recovery using the traditional method of recovery based on image copies and archive logs.

Be sure to have all of the information needed for the recovery. The required image copies of all the data objects will be the same, but now all the bootstrap data sets (BSDSs) and archive logs from all members must be provided using one of three options:

- **Archive log mode(quiesce)**

As previously explained, this command enforces a consistency point by draining new units of recovery. Therefore, this command is restrictive for providing continuous availability but, under successful execution, it gets a group-wide point of consistency whose log record sequence number (LRSN) is specified in the BSDS of the triggering member.

- **Archive log mode(group)**

With this command, members of the group are not quiesced in order to establish a point of consistency, but all of them register a checkpoint for their log offload. Because you are going to conditionally restart all the members of the group, you must find a common point in time on the log in order to provide for consistency throughout the group. You must find the lowest ENDLRSN of all the archive logs generated (see message DSNJ003I), subtract 1 from the lowest LRSN, and prepare the conditional restart for all members using that value.

- **Set log suspend**

If you plan to use a fast volume copy of the system, remember that the suspend command does not have group scope, so that it must be triggered in all group members before splitting pairs or performing FlashCopy.

At the recovery site, remember that each member's BSDS data sets and logs are available. The logs and conditional restart must be defined for each member in the respective BSDS data sets. The conditional restart LRSN for each member must be the same. Contrary to the logs and BSDS data sets, the DB2 Catalog and Directory

databases exist only once in the data sharing group and must only be defined and recovered once from any of the active members.

DSNJU004 and DSN1LOGP have options that allow for a complete output from all members.

After all members are successfully restarted and if you are going to run single-system data sharing at the recovery site, stop all members except one by using the STOP DB2 command with MODE(QUIESCE). If you planned to use the light mode when starting the DB2 group, add the LIGHT parameter to the START command. Start the members that run in LIGHT(NO) mode first, followed by the LIGHT(YES) members.

You can continue with all of the steps described in topic *Performing remote site recovery from a disaster at a local site* in *DB2 for z/OS Administration Guide* (SC19-4050).

Tracker site for disaster recovery

A DB2 tracker site is a separate DB2 subsystem or data sharing group that exists solely for the purpose of keeping shadow copies of your primary site data.

No independent work can be run on the tracker site. From the primary site, you transfer the BSDS and the archive logs, then the tracker site runs periodic LOGONLY recoveries to keep the shadow data up-to-date. If a disaster occurs at the primary site, the tracker site becomes the takeover site. Because the tracker site has been shadowing the activity on the primary site, you do not have to constantly ship image copies. The takeover time for the tracker site can be faster because DB2 recovery does not have to use image copies.

The general approach for tracker site recovery based on the DB2 *BACKUP SYSTEM* is as follows:

1. Use *BACKUP SYSTEM* to establish a tracker site.
2. Periodically send active, bootstrap data set (BSDS), and archive logs to tracker site (Metro Mirror, Global Mirror, z/OS Global Mirror, FTP, or tapes).
3. Send image copies after load/reorg log(no).
4. For each tracker recovery cycle:
 - Run RESTORE SYSTEM LOGONLY to roll database forward using logs.
 - Use image copies to recover objects that are in recover pending state.
 - Rebuild indexes that are in rebuild pending state.

More information about setting up a tracker site and recovery procedures can be found in the *DB2 for z/OS Administration Guide* (SC19-4050) and *DB2 for z/OS Data Sharing: Planning and Administration* (SC19-2973).

Chapter 4. Network characteristics for high availability

Read about high availability aspects of the network between a remote SAP application server and the SAP database server. In the solution that is described in this publication, the network is between an SAP application server on a non-z/OS operating system and the SAP database server on z/OS. It shows how highly available network connections can be set up in between.

First, some general recommendations for high availability are presented in this topic. Then, the three recovery mechanisms that are needed to avoid outages due to network component failures are explained. Based on these mechanisms, the recommended network setup is developed, supported by the experience from the test team. For the sample definitions of this test scenario, see “Network setup” on page 293. These sample definitions give you an impression of the necessary implementation tasks.

The last part of this topic concludes with a description of an alternative recovery mechanism, which is implemented by the intraensemble data network (IEDN) of zEnterprise and zBX. It informs about the requirement to use z/OS VIPAs, and ends with a description of timeout behavior.

Network considerations

Read this information to obtain some basic knowledge about networks that help you understand the measurements to be taken to ensure a high availability of the components within the mentioned network.

A communications network can be subdivided into a physical communication layer and a software communication layer. The physical layer can be broken down into the network infrastructure (cabling, active components such as hubs, switches, and routers) and the network interface card (NIC). The software layer comprises, for example, the TCP/IP stack, the device driver, and the microcode.

Virtualization of physical resources adds an intermediate layer. When running Linux under z/VM[®], many of the networking resources may be virtual ones. z/VM offers virtual network switches, LANs, and NICs in addition to the virtualization of existing network devices (OSAs, HiperSockets).

Planned or unplanned outages of a network result in interruptions of the communication path between the remote application server and the z/OS database server. If no recovery mechanism is in place, this results in a direct service interruption for the end users.

The impact levels of network failures can be classified according to their impact on the SAP end user:

Transparent or no impact

This is the most desirable level.

Reconnect

The user interface is blocked until the SAP application server has reconnected to an IBM z Systems DB server. All running transactions are rolled back (the user may have to re-enter data).

New logon

Active users have to log on to the SAP system again.

Downtime

No logon is possible. This is the least desirable level.

If you set up *DB connection failover* correctly, all network outages can be recovered with it. However, DB2 connection failover always performs at least one reconnect, which means that it cannot be used to implement the most desirable level of user impact, the transparent level.

TCP/IP implementations under z/OS, AIX, and Linux on z Systems, also offer fault-tolerant features to recover from network and NIC failures, for example. These recovery mechanisms are:

- Dynamic routing of the IP layer based upon the Open Shortest Path First (OSPF) routing protocol
- Virtual IP Addresses (VIPAs)

To attain the transparent level of recovery from network failure, it is necessary to fully utilize the communication network recovery options available in all of the communication network layers (real and virtual). Individually each communication layer offers some level of recovery, but it is the correct combination of recovery features that lead to transparent recovery.

This section provides hints and recommendations on how to achieve the transparent level of recovery on z/OS, AIX, and Linux on z Systems, utilizing communication network options, such as:

- HiperSockets – z Systems Internal LAN
- OSPF – IP routing via the Open Shortest Path First protocol
- PMTU Discovery – Path Maximum Transmission Unit Discovery
- VIPA - Virtual IP address, including the Source VIPA feature and methods
- VLANID – IEEE 802.1Q virtual LAN identifier
- VSWITCH – z/VM Virtual Switch

After some general recommendations, all three recovery mechanisms (DB2 connection failover, OSPF, and VIPA) are explained in detail. Then our recommended high availability network setup is introduced. It provides transparent recovery of most kinds of network outages. Note that if you do not have such requirements of your network availability, and if you are willing to take the risk that a physical switch outage may mean a network outage and therefore an SAP outage, read “Alternative network setup” on page 54, which describes a much simpler setup.

General recommendations

In various subtopics, this topic describes considerations and recommendations to keep in mind when setting up a high availability network.

- “Hardware considerations” on page 41
- “z/OS communication software considerations” on page 41
- “Considerations for the Linux on z Systems application server” on page 42
- “Multiple Linux on z Systems guests under z/VM” on page 42
- “Considerations for the AIX application server” on page 44

Hardware considerations

In a highly available network, there must be no single point of failure. At a minimum, this means the duplication of all network components of the physical layer, such as network adapters, network control equipment, switches, and cables. Do not confuse no single point of failure with no failure. Design your network with the assumption that every single component will fail. Note that high mean-time-between-failure (MTBF) does not mean a component will function correctly for that time. Dual parts such as power supplies etc. in a single component, while good to have, can never match the total duplication of that component. Buying the most expensive components with guarantees of stability is tempting, but often duplication of alternative hardware components offers no single point of failure at less or at a similar cost. Duplication often makes planned component outages, for routine maintenance etc., that much easier.

Ensure as much redundancy as possible with regard to power and cooling. If a power failure or a cooling system outage can stop your duplicated components at the same time then you have failed to achieve your goal. Use intelligent network components that can be monitored, for example with SNMP, then monitor your network components and recover from failures quickly. Once a network component fails, until it is fixed or replaced, you are running with a single point of failure if you only have one backup component.

With duplicated network components you have at least two totally different and independent physical network paths to the z/OS database server from each remote application server. This is the case with the IEDN of zEnterprise and zBX.

If the application server and DB server are both running on the same z Systems CEC, then the best network performance will be gained by utilizing HiperSockets for the network paths. But if you have multiple z Systems CECs then in order to plan for scheduled/unscheduled outages include standard LAN network paths in addition to the HiperSockets.

To obtain optimum network performance for remote application servers connected via a LAN, use switched OSA-Express® Gigabit Ethernet or faster, and exploit jumbo frames with an MTU of 8992. This has superior latency and capacity.

Note: For a network connection to utilize Jumbo frames correctly between two hosts, Jumbo frames must be enabled on both of the hosts and on the associated physical switch ports of each host.

z/OS communication software considerations

It is recommended that you have only one AF_INET TCP/IP (INET) stack defined, the Integrated Sockets AF_INET stack. In addition to the overhead that is intrinsic to the Common AF_INET (CINET) stack, defining more than one TCP/IP stack by including the Common AF_INET stack can complicate setup and operations considerably.

Note:

1. Because Path MTU Discovery is switched off by default under z/OS, you need to use the PATHMTUDISCOVERY keyword in the IPCONFIG statement of your TCP/IP profile to indicate to TCP/IP that it should dynamically discover the path MTU, which is the minimum MTU for all hops in the path. Enabling

Path MTU Discovery can avoid IP segmentation which can be time-, memory-, and CPU-intensive especially when IP packets arrive over multiple network paths and out of sequence.

2. For further guidelines that might apply when activating Path MTU discovery, refer to *z/OS Communications Server IP Configuration Guide*.

Settings for optimal GbE performance

As mentioned in “Hardware considerations” on page 41, it is recommended that you use jumbo frames with MTU of 8992. Set the TCP send and receive buffer sizes to at least 64 KB. Starting with z/OS V2R1, this is the default. In installations with z/OS earlier than V2R1, set the following parameters in the TCPIP profile:

```
TCPCONFIG TCPRCVB 65536 TCPSENDB 65536
```

DELAYACK must be active with TCPIP on z/OS, this is the default.

Considerations for the Linux on z Systems application server

If a Linux on z Systems application server runs in one LPAR and the SAP on DB2 database server runs in another LPAR within a single z Systems server, HiperSockets™ is the preferred method of connectivity because of the superior performance characteristics of HiperSockets as compared to all other modes of LPAR-to-LPAR communication.

Check for the latest networking restrictions relating to your combination of z/VM release, z Systems hardware model, and OSA card under **Networking** at:

http://www.ibm.com/developerworks/linux/linux390/development_restrictions.html

Make sure you have read *SAP Note 1263782: DB2-z/OS: Recommended settings for HiperSockets (zLinux)*. This SAP Note describes the recommended settings for HiperSockets-communication between SAP application servers running under Linux on z Systems and z/OS DB2 Database Server and/or SAP Enqueue Server.

Settings for optimal GbE performance

As mentioned in “Hardware considerations” on page 41, it is recommended that you use jumbo frames with MTU of 8992. It is recommended that you use at least 64 KB as the TCP send and receive buffer sizes. Be aware that the default value for Linux is 16 KB. It is recommended that you switch off rfc1323, if you run with buffer sizes of 64KB.

Under Linux, option `tcp_window_scaling` is used for this purpose, and by default it is switched ON. It is recommended to switch it to OFF because it adds 12 bytes of overhead to the TCP header. You can change the values by adding them to `/etc/sysctl.conf`:

```
net.ipv4.tcp_wmem= 4096 65536 131072
net.ipv4.tcp_window_scaling = 0
```

For more information, refer to *SAP Note 1557416: DB2-z/OS: Recommended settings for Gigabit Ethernet*.

Multiple Linux on z Systems guests under z/VM

If you are running several Linux on z Systems guests (as SAP application servers) under z/VM, it is recommended that you set up an internal virtual LAN for the

Guests that is based on z/VM Virtual Switch (VSWITCH) technology. External network connectivity for any z/VM Guest on z Systems will always be via an OSA card. Rather than having each Guest managing its own OSA, or directly sharing one, z/VM takes control of the OSA port (CHPID) and virtualizes an OSA port to each connected Guest. To a Guest connected to a VSWITCH, it appears to be a regular OSA port and therefore no specialized driver support is required. Each Guest on a VSWITCH communicates with other similarly connected Guests or external hosts connected via the VSWITCH OSA port as if they were all on a local LAN, no routing of IP traffic is required. If the OSA port becomes unavailable Guests can continue to communicate with each other on the same VSWITCH, though not with external hosts.

z/VM also provides VSWITCH failover, which offers redundancy within the same IP subnet. The VSWITCH must be connected to two OSA ports, and those in turn with separate physical switches which are connected through a trunk line and configured to create one common LAN segment and hence IP subnet. Failure of an OSA card/port or physical switch can then be recovered at the z/VM level. Failures that are handled and recovered by z/VM are transparent to the Guest. z/VM itself does not handle TCP traffic, so any disruption on the TCP layer which is not resulting in a error indication by the OSA card or switch (for example, erroneously deactivated switch port) needs to be handled by other means, for example OSPF hello packets. If z/VM does not detect an error then no VSWITCH failover takes place.

A z/VM VSWITCH virtual LAN may optionally use IEEE 802.1Q VLANs. VLANs have become a popular method of logically connecting many hosts onto the same IP subnet even though they may be physically dispersed throughout a company's local network fabric (this avoids the use of routers and gateways for local traffic). One such example could be local SAP GUI end users which you may wish to connect to the SAP applications servers running on Linux on z Systems. If you run multiple SAP instances you may wish to keep the traffic of each SAP instance limited to a specific VLANID. Therefore rather than having a separate OSA port for each VLAN and specifying the VLANID on the associated switch ports, you can have a single OSA port specified on the switch as a TRUNK port, and then specify the specific VLANIDs on the VSWITCH instead.

Another z/VM VSWITCH option is IEEE 802.3ad link aggregation, which allows multiple OSA ports to be connected to a physical switch that also supports IEEE 802.3ad link aggregation. Link aggregation allows for a higher combined throughput, and individual link redundancy.

Figure 9 on page 44 depicts the use of VSWITCH.

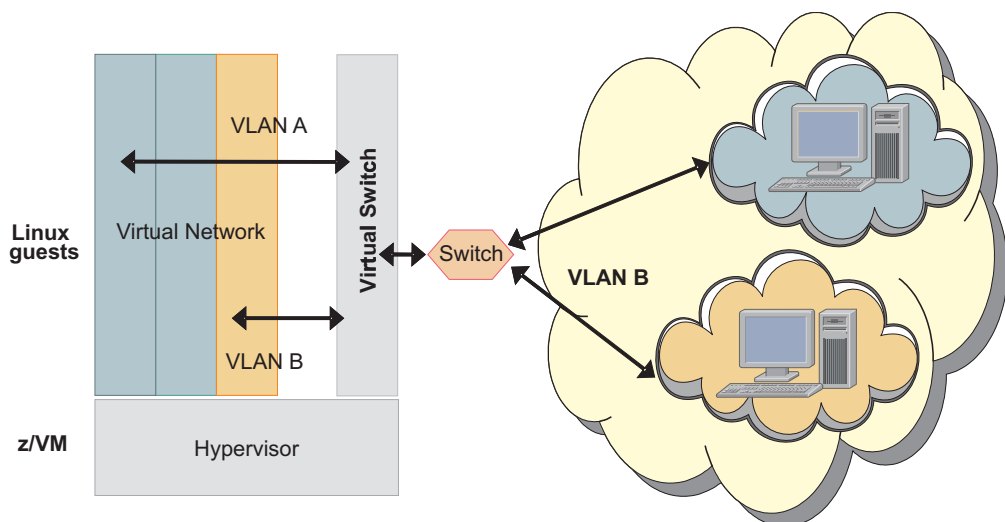


Figure 9. Sample VSWITCH utilization

For a detailed description of how the mentioned features can be utilized by Linux guests, and how your virtual networking configurations can be greatly simplified through the use of these functions, read the IBM Redpaper™ *Linux on IBM eServer zSeries and S/390: VSWITCH and VLAN Features of z/VM 4.4*, REDP-3719.

This Redpaper also contains a section entitled *High availability using z/VM Virtual Switch*, which describes what is needed to use VSWITCH technology to create highly-available connectivity for your Linux guests under z/VM. You configure the redundancy features of VSWITCH and combine them with LAN-based high availability features. You define multiple OSA-Express adapters for hardware redundancy, and multiple TCP/IP controller service machines for some software redundancy. As long as your LAN switch is configured appropriately, you can ensure that your z/VM guests stay linked to the external network when failures occur.

Considerations for the AIX application server

You should be aware of the following Path MTU discovery and performance considerations for the AIX application server.

Path MTU discovery

It is recommended that you use Path MTU discovery. Since AIX 6.1, Path MTU discovery is ON by default and it can be used on duplicate routes. You can use multiple routes to the same destination (including multiple default routes) through the Multipath Routing feature. The cost (hop count) is used to determine the route to use when there are multiple routes to the same destination.

By default, a round-robin scheme is used to select a route when there are multiple routes with the *same destination and cost* (equal cost routes). In addition, different multipath routing methods can be defined via the SMIT `mkroute` fast path.

Settings for optimal GbE performance

As mentioned in "Hardware considerations" on page 41, it is recommended that you use jumbo frames with MTU of 8992.

It is also recommended that you use at least 64 KB as the TCP send and receive buffer sizes. Check that the actual values are at least 64 KB for adapters that can have a much higher send and receive space than 64 KB, but be aware that the default for z/VM is 16 KB.

You should furthermore switch off rfc1323, if you run with buffer sizes of 64KB. Since z/VM 6.1, rfc1323 is OFF by default, because rfc1323 is not required to achieve maximum network throughput on local networks with 10/100/1000 Ethernet. Even if it can be enabled for an adapter, it is recommended that you switch it OFF because it adds 12 bytes of overhead to the TCP header.

For more information, refer to *SAP Note 1557416: DB2-z/OS: Recommended settings for Gigabit Ethernet*.

DB2 connection failover recovery mechanism

Read this information to get a starting point for implementing a setup for high available database connections for SAP on DB2 for z/OS.

DB2 connection failover is the mechanism by which the database connection of SAP ABAP or Java instances can be redirected to a standby database server in case the primary database server becomes inaccessible. Exploiting DB2 data-sharing in a sysplex, you can thereby provide redundancy at the database service layer. Also refer to “DB2 connection failover” on page 15.

Together, both features (DB2 connection failover and DB2 data sharing) address failures of, for example, the database server, the network, and z/OS. When an SAP work process detects that its primary database server has become inaccessible, it rolls back the current SAP transaction and automatically reconnects to the standby DB server. When the primary DB server is back up or the standby DB server becomes inaccessible, it is possible to switch back to the primary DB server.

To implement the solution in “Recommended setup for high availability connections between client and server” on page 48, the following preconditions must be met:

- DB2 data sharing must be set up and the primary and standby database servers must be members of the same data sharing group.
- All network components need to be duplicated.
- DB2 connection failover is set up correctly for ABAP and Java instances (see “DB2 connection failover” on page 15).

It is possible to define different system configurations to handle the failure of one or several components. In the configuration shown in Figure 11 on page 52, each DB2 data sharing member runs in a separate LPAR on a separate sysplex machine and serves as primary database server for one application server and as standby database server for another.

OSPF protocol as a recovery mechanism

Read this topic, if you want to use the OSPF protocol as a recovery mechanism (instead of the DB2 connection failover recovery mechanism) because of the mentioned advantages.

Open Shortest Path First (OSPF) is a dynamic link-state routing protocol. It aids recovery of TCP/IP connections from network failures by finding an alternative

path to the destination. The IP layer then uses this path to actually route IP packets to the destination. Compared to other routing protocols, OSPF updates its routing table faster and has a shorter convergence time.

OSPF itself is able to quickly detect topological changes in the network by sending small packets to test neighbor routers and links. In addition, it reacts to failures discovered by the TCP/IP stack or hardware components rapidly. For example, when a channel detects an error under z/OS, which usually happens within milliseconds, OSPF can update its routing table almost immediately, at the latest after OSPF's *dead router interval*, which is 40 seconds by default.

Then it sends small Link State Advertisements (LSA) to its peers in order to trigger a recalculation of their routing tables. The peers recalculate their routing tables usually within milliseconds. This short convergence time is one advantage over other routing protocols. When TCP automatically re-sends data that was not acknowledged because of a network failure, the data automatically uses the new routing table entry and the alternate path.

In order to have an alternative *physical* path to a destination, all network components must be duplicated.

OSPF calculates the cost for a path by calculating the sum of the costs for the different links in the path. The cost for a link is derived from the interface bandwidth of that link. That cost has to be configured for each link. For example, you can configure the cost for a Gigabit Ethernet link as 15 and for a Fast Ethernet link as 30. Correctly configuring the costs is critical for establishing the desired routes and may vary in different networks. In general, choosing the routes with the least-cost path can be achieved by configuring the cost inversely proportional to the bandwidth of the associated physical subnetworks.

Additionally, OSPF supports Equal Cost Multipaths under z/OS, AIX, and Linux on z Systems. These are parallel paths to a destination which all have the same cost.

The OSPF routing protocol is implemented by:

- the OMPROUTE daemon under z/OS
- the gated daemon under AIX
- the quagga and ospfd daemons under Linux on z Systems

Note: In terms of Linux, quagga is a derivative of zebra.

For general information on dynamic routing with OSPF on z/OS, see the *z/OS Communications Server IP Configuration Guide*.

Virtual IP Address (VIPA) as a recovery mechanism

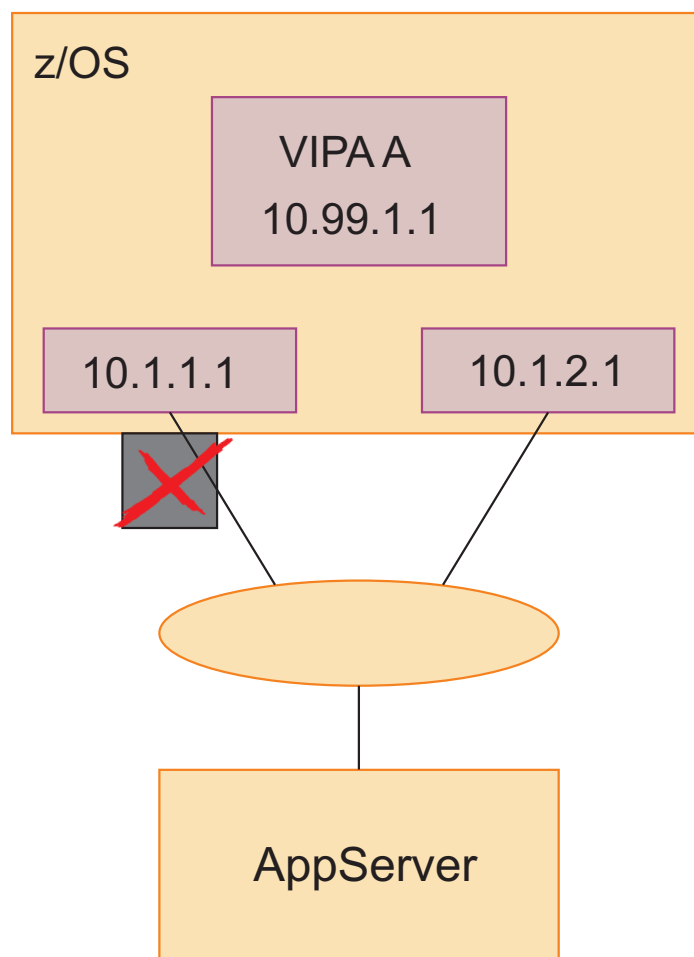
Read this topic for information on how to avoid an end point problem in a TCP/IP network using a Virtual IP Address (VIPA).

In a TCP/IP network there exists the so-called *end point problem* of a TCP/IP connection. A normal, unique IP address is associated with exactly one physical network interface card (NIC). If the NIC fails, the IP address is no longer reachable. If the IP address of the failed NIC is either the source or the destination

of a TCP/IP connection, it is not possible to route around it. Therefore, an *end point NIC* is a single point of failure (SPOF). A Virtual IP Address (VIPA) solves this end point problem.

A VIPA is an IP address that is associated with a TCP/IP stack and is not tied to a physical interface. It is therefore less likely to fail. It can be reached via any of the physical interfaces of that TCP/IP stack and it is advertised to the IP routers via dynamic routing. Therefore, if one of the NICs fails, the VIPA can still be reached via one of the other NICs and a NIC is no longer a SPOF.

You must run a dynamic routing protocol like OSPF when exploiting VIPAs. You must also define a subnet for the VIPAs, different from the subnets of the other IP addresses of the NICs. Figure 10 illustrates how a VIPA and OSPF work together under z/OS to achieve transparent recoveries from z/OS device or NIC (feature) failures:



SAPDBHOST = 10.99.1.1

Figure 10. VIPA and OSPF recovery mechanisms under z/OS

The VIPA A (10.99.1.1), which belongs to subnet 10.99.1, represents the z/OS application (DDF instance, NFS, or SCS) to the client. Initially, the traffic to the VIPA flows via the NIC with IP address 10.1.1.1, which belongs to the 10.1.1 subnet. When this NIC fails, OSPF on z/OS detects the failure, finds the alternate path to the VIPA subnet (10.99.1) via the 10.1.2 subnet, and updates the local

routing table. OSPF advertises the change to its peers via LSAs. The peers recalculate their routing tables. Subsequently, the traffic to the VIPA flows via the NIC with IP 10.1.2.1.

For transparent recoveries from NIC failures on the non-z/OS application server side, an additional functionality of VIPAs, the so-called Source VIPA function, must be exploited because the SAP work processes are the initiators of the connections to the database server (see “VIPA and source VIPA functions on remote application servers” on page 49 for details).

VIPAs are supported on z/OS, AIX, and Linux on z Systems. VIPAs on AIX are always Source VIPAs. For information on alternative recovery mechanisms on Windows, see “Alternative network setup” on page 54.

On z/OS, two different types of VIPAs are supported: *static* VIPAs and *dynamic* VIPAs. Both are equally capable of aiding recovery from end point failures such as the one described in this scenario. It is recommended that you use static VIPAs for database connections, whereas dynamic VIPAs should be used for movable applications like the NFS server and SAP Central Services.

You may also use an own dynamic VIPA for a DB2 data sharing member. However, this is not supported by the standard **DB2 add-on policy*.

If you want to use dynamic VIPAs and move your DB2 data sharing members between LPARs, then you must adapt your SA z/OS policy and make sure that the dynamic VIPA is started together with the DB2 data sharing member on the correct LPAR.

For general information on the z/OS VIPA function, see the *z/OS Communications Server IP Configuration Guide*.

Recommended setup for high availability connections between client and server

This section describes the recommended setup aspects for OSPF and subnet configuration and for VIPA and source VIPA functions on remote application servers.

The following topics are handled:

- “OSPF and subnet configuration aspects”
- “VIPA and source VIPA functions on remote application servers” on page 49
- “Recommended setup for a high availability network” on page 51
- “Additional considerations” on page 53

OSPF and subnet configuration aspects

In an SAP on z Systems environment, transparent recoveries from NIC failures with OSPF can only be achieved if:

- All NICs on a machine belong to different subnets and
- VIPAs are set up on all machines in the system, on the database servers as well as on the application servers.

A host in a subnet is either directly accessible in its local subnet or it is in a remote subnet and the first gateway in the path to that subnet is directly accessible. OSPF

does not change a subnet route if a host in a directly accessible subnet becomes inaccessible but other hosts in the subnet are still accessible.

OSPF changes a route to a subnet only in the following two cases:

- Case A, where both OSA adapters/NICs are in the same subnet: If OSPF's own primary NIC connecting to a directly accessible subnet fails, it switches the route to that subnet to the backup (secondary) NIC. For OSPF, the primary NIC connecting to a subnet is the adapter which is used to exchange OSPF data. If the secondary NIC fails, OSPF will not have to change its current route to that subnet as OSPF can still happily talk to the subnet over its primary NIC. However, in a 'one subnet' environment with VIPA support and two separate connection paths, OSPF's primary NIC may not be the NIC over which the SAP database traffic flows:

The problem can be solved if OSPF recognizes each adapter on a machine as its primary NIC to a subnet. This can be achieved by running each NIC on a machine in its own subnet.

- Case B, where both OSA adapters/NICs are in the same subnet: OSPF recalculates the route to a subnet/host which is not directly accessible ('remote'), if its 'gateway' to the remote subnet/host is down.

Consequently, if the NIC on a non-z/OS application server fails, OSPF on z/OS does not recalculate its routing table, because the directly accessible subnet, to which the failed NIC belongs, is still reachable (case A) and this subnet has no gateway to another remote subnet.

On the application server, however, OSPF does recalculate the route for the outbound traffic to the z/OS VIPA subnet, because its gateway to the remote z/OS VIPA subnet has failed. As a result, the routing tables on the two sides differ and the users connected to this application server experiences a downtime.

The problem can be solved where a remote subnet/host becomes inaccessible when the NIC on the application server fails. This can be achieved by defining a VIPA on the non-z/OS application server. Then, OSPF on z/OS will also recalculate its routing table and the routing tables will converge.

For the configuration shown in Figure 11 on page 52, this means, that six different subnets are needed to exploit VIPA on both sides, on the z/OS database server and on the applications servers on AIX and Linux on z Systems.

Optionally you can run with four subnets only, with two subnets for the OSAs, one for the z/OS VIPAs and one for the remote application server VIPAs, if you define the VIPAs as hosts (/32 bit mask) to OSPF.

VIPA and source VIPA functions on remote application servers

Due to the fact that each SAP work process on an application server initiates a TCP/IP connection to the z/OS database server and due to the way TCP/IP handles connection establishment etc., an additional feature of VIPAs, the so-called Source VIPA function, is needed on the application server side:

- Without Source VIPA: When the Source VIPA function is not used and a request to set up a connection is processed on the application server, the IP address of the NIC of the application server is put into the 'request' IP packet as source IP address before it is sent to z/OS. z/OS sends its response to exactly that source IP address. This behavior does not allow the exploitation of VIPAs on the application server side, because this means that – viewed from the z/OS side – the application server VIPA never shows up as the IP address of a connection

that 'originates' on the application server. This makes transparent recoveries from adapter failures on the application server impossible.

- With Source VIPA: When the Source VIPA function is used, the VIPA is put into the IP header of an IP packet as source IP address, and the exploitation of VIPA on the application server allows transparent recoveries from NIC failures on the application server.

The VIPA function is available in AIX. The administrator can control for which interface(s) the VIPA is used as source address for outgoing packets (source VIPA).

The VIPA function is available on Linux on z Systems via the so-called dummy device. For detailed information concerning the definition of a VIPA under Linux on z Systems, see *VIPA - minimize outage due to adapter failure in Linux on z Systems - Device Drivers, Features and Commands*, SC33-8411, available from

http://www.ibm.com/developerworks/linux/linux390/documentation_dev.html

It is not recommended that you use the Source VIPA utility as described because of its dependency on the LD_PRELOAD feature which for security reasons is disabled for any processes running with UID=0.

In the Device Drivers manual, the section on Standard VIPA is relevant. Of special importance is the `qethconf` command which must be used to register any Linux VIPAs into any OSA ports that are used as gateways for the VIPAs on the local interfaces. For example if you have a `dummy0` interface with a VIPA of `10.1.100.1` and two local OSA interfaces `eth0` and `eth1`, then the following commands must be issued:

```
qethconf vipa add 10.1.100.1 eth0
qethconf vipa add 10.1.100.1 eth1
```

Failure to issue the two shown commands results in inbound IP packets with a destination-IP-address of the VIPA `10.1.100.1` being dropped by the OSA card(s). This is because the OSA card is operating at the Layer 3 level and supports multiple IP addresses with a single MAC address. If the VIPA is not registered, the OSA does not know the device numbers to which the IP packet should be forwarded. See "Static VIPA definitions required for SUSE" on page 300 for a solution about how to issue the `qethconf` command when an `eth<x>` interface is displayed at boot time (in this solution the `setvipa` script was used).

Quagga on SLES 11 and RHEL 5.1 or higher now provides the ability to set the source IP entry in any routes that it adds to the IP stacks routing table.

All routes that are learned first by the **ospfd** daemon are passed to the **zebra** daemon which can process them before passing them to the IP stack via the NETLINK interface.

The **zebra** daemon has a well established route-map and prefix list filter feature, to which has now been added the ability to set a source IP via a **set src** sub-command.

For this example, assume that all z/OS VIPAs are in the subnet `10.1.100.0/24`, and that you only want to set the source IP of our own VIPA address `10.1.200.1` for routes to these z/OS VIPAs.

The following statements need to be added to `/etc/quagga/zebra.conf`:

```
route-map vipa1 permit 10
match ip address prefix-list DEST
set src 10.1.200.1
continue
route-map vipa1 permit 20
ip protocol ospf route-map vipa1
ip prefix-list DEST permit 10.1.100.0/24 le 32
```

Recommended setup for a high availability network

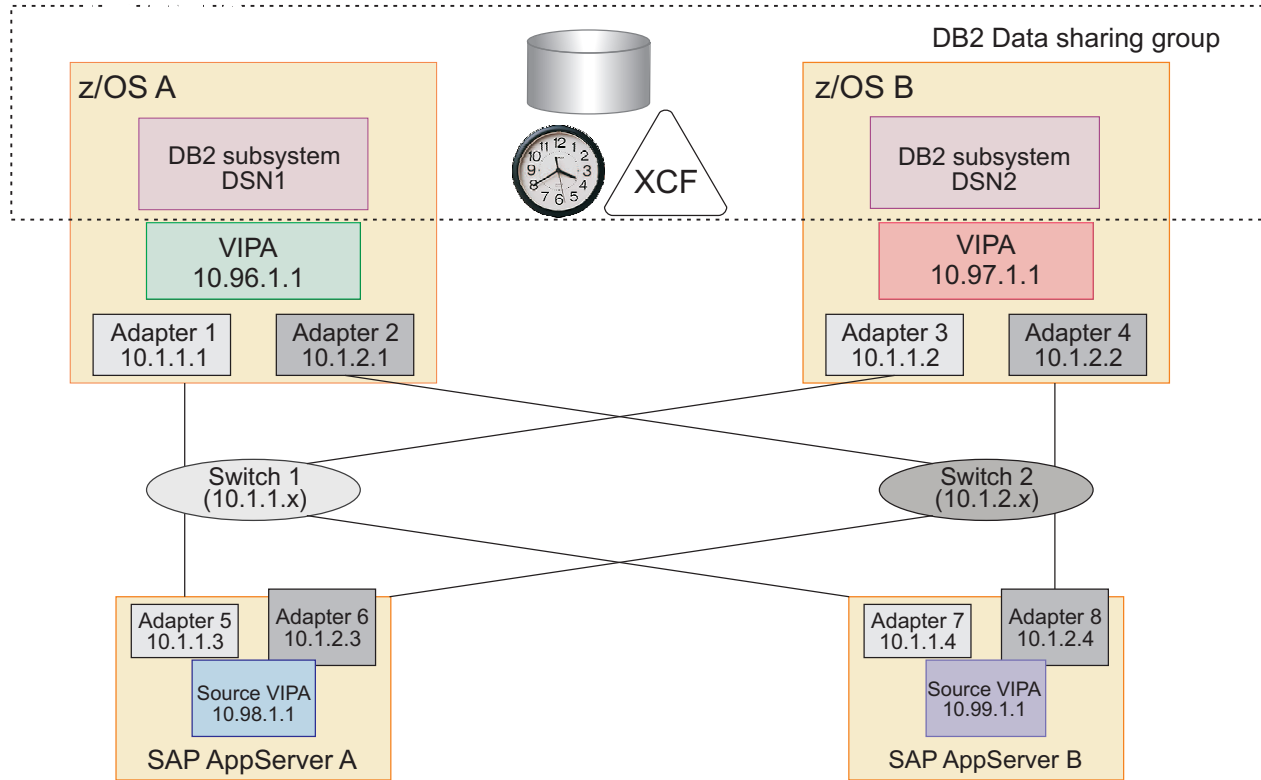
Figure 11 on page 52 shows the recommended setup for a high availability network between the SAP application server and the z/OS database server (or NFS, SCS, and so on) that results from the previous considerations in this topic:

- DB2 data sharing (for DB server)
- Duplicate network hardware components
- DB2 connection failover (for ABAP and Java application servers)
- Different subnets for OSPF
- VIPA exploitation on z/OS
- VIPA and Source VIPA exploitation on the application server side.

Important:

This recommended HA network setup allows **transparent** recovery of most kinds of network outages. If you use OSPF, any failure in the network path is detected and can be handled transparently to the highest degree, this is because failing OSPF heartbeats which probe the network path. If you do not have such demanding requirements for your network availability, read “Alternative network setup” on page 54, which describes a much simpler setup.

The intraensemble data network (IEDN) of zEnterprise and zBX is an implementation of this alternative network setup.



: Each color represents a subnet

Figure 11. Recommended setup for a high availability network

In this configuration, all NICs on one machine (z/OS and remote application server) and all VIPAs belong to different subnets. This generates the following routing alternatives:

- VIPA 10.96.1.1 (of subnet 10.96.1.x) on z/OS A can be reached from SAP application server A by normal IP routing over subnet 10.1.1.x (10.1.1.3 - Switch 1 - 10.1.1.1) or subnet 10.1.2.x (10.1.2.3 - Switch 2 - 10.1.2.1).
- Source VIPA 10.98.1.1 (of subnet 10.98.1.x) on SAP application server A can be reached from z/OS A by normal IP routing over subnet 10.1.1.x (10.1.1.1 - Switch 1 - 10.1.1.3) or subnet 10.1.2.x (10.1.2.1 - Switch 2 - 10.1.2.3), accordingly.

Alternatively, you can run with four subnets only, if you define the VIPAs as hosts (/32 bit mask) to OSPF. Two subnets for the OSAs, one for the z/OS static VIPAs and the remote application server VIPAs and one for the dynamic z/OS VIPAs.

The following table shows the recovery attributes of the recommended setup.

Table 2. Recovery attributes of the recommended setup

Failing network component	Recovery mechanism	Impact on SAP end users
NIC on application server	OSPF/VIPA	Transparent
NIC on z/OS, switch, cable	OSPF/VIPA	Transparent
z/OS TCP/IP stack	DB2 connection failover	Reconnect (directly or after one connect timeout)

The remote application server detects the failure of the switch not later than the end of the OSPF's *dead router interval*, which is 40 seconds by default. If a shorter interval is required, it is recommended to use a value of 10 seconds (or a different value which fits your requirements after careful investigation).

Additional considerations

Assume that the primary DB server (static VIPA 10.96.1.1) of application server (AS) A is the data-sharing member DSN1 running in the z/OS A host, and its secondary DB server (static VIPA 10.97.1.1) is the z/OS B host running DSN2. Also assume that AS A's DB connections go to DSN1.

If the z/OS A host is down or not reachable, the AS A running OSPF quickly detects that there is no longer a route to the VIPA 10.96.1.1. Re-connection requests initiated by the application server DB2 connection failover functionality are routed or forwarded to the default gateway of the AS. If the z/OS VIPA belongs to a private network, such as the 10.x.x.x, the gateway does not forward such a packet, it deletes it and sends a connection refused reply.

Because the DB2 connection failover functionality does three retries to the VIPA 10.96.1.1 of its primary DB server, it quickly receives three connection refused replies and almost immediately tries to connect to the VIPA 10.97.1.1 of its secondary DB server, the z/OS B host (which runs the DSN2 data-sharing member). This behavior has no negative impact on System Automation for Multiplatforms behavior.

A failover to the secondary DB server does *not* happen immediately, it takes minutes with the TCP default settings, if the static VIPA of z/OS A belongs to a subnet which is forward-able by the default gateway. This means that if the connect IP packet is not dropped by the default gateway it does not generate a 'connection refused' reply.

Because there is no reply, the AS re-connection attempt has a connection timeout which is per default about 75 seconds on AIX and 180 under Linux. The **DB2 connection failover** functionality starts three retries until it eventually tries to connect to the static VIPA of the secondary DB server. The total time for a switch is too long for System Automation for Multiplatforms, which results in the AS being stopped.

If you cannot implement a setup where the default gateway drops the connection request packet, you must adapt the following TCP parameter to achieve an acceptable time for application server *DB2 connection failover*:

SUSE Linux Enterprise Server releases supported by Linux on z Systems:

```
sysctl -w net.ipv4.tcp_syn_retries=2
```

or adapt the setting permanently in:

```
/etc/sysctl.conf: net.ipv4.tcp_syn_retries=2
```

AIX supported releases:

```
no -p -o tcp_keepinit=40
```

make the setting permanent with an entry in `/etc/tunables/nextboot`.

Note: The described settings are valid for all established outgoing TCP V4 connections on the application server system.

An extended failover time is also observed if OMPROUTE under z/OS is set up in a way to advertise a subnet route for the static VIPA to its neighbors and the static VIPA fails but the TCP stack remains operational, which is a very unlikely scenario.

Alternative network setup

This topic describes a simplified network configuration with less complexity. The configuration does not use a dynamic routing protocol and multiple subnets. All network adapters on z/OS and on the SAP application server platform plus network switches belong to the same network or subnet, which significantly simplifies the setup. The configuration provides less availability in some failure scenarios, which is discussed in a separate topic. The intraensemble data network (IEDN) of zEnterprise and zBX is an implementation of this alternative network setup.

The alternative network setup applies to environments with DB2 and SAP central services on z/OS and SAP application servers on Linux, AIX or Windows. For adapter redundancy on Linux, AIX, or Windows you could use, for example, EtherChannel or Virtual Input Output server (VIO) under AIX, channel bonding under Linux, and the **adapter teaming** function of Windows. The following sample shows a z/OS / AIX environment.

Such a configuration is also discussed in the following IBM z/OS Newsletter in section: *No dynamic routing protocol? No problem!*

<http://publibz.boulder.ibm.com/zoslib/pdf/e0z2n161.pdf>

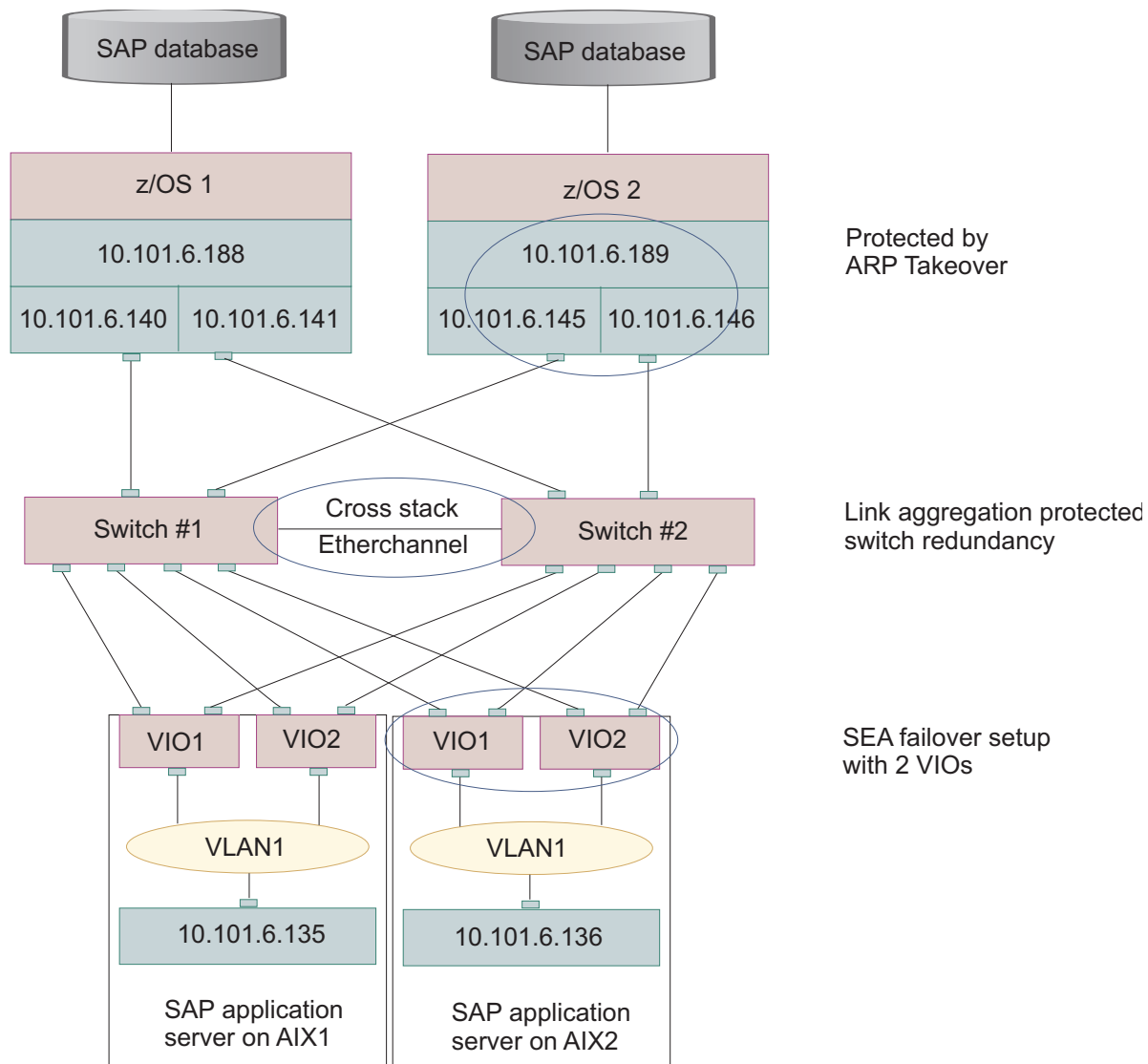


Figure 12. Alternative high availability network configuration

To achieve an acceptable level of network high availability three areas need to be secured by eliminating single point of failures:

1. z/OS network adapter
2. Network switch
3. AIX network adapter

How to secure these areas is described in the following sections.

z/OS network adapter

On the z/OS side this configuration uses Address Resolution Protocol (ARP) takeover to transparently handle the failure of one OSA-Express feature in a redundant setup of at least two physical interfaces.

Usage of ARP takeover: ARP takeover is a function that allows traffic to be redirected from a failing OSA connection to another OSA connection. This function is supported with IPv4 and IPv6 OSA interfaces. ARP takeover uses LAN

broadcast protocols. Therefore, to use ARP takeover, all of the z/OS systems that belong to your SAP installation must be in the same LAN respectively in the same TCP/IP subnet. This is called a flat network topology. No bridging, no routing, all participants are within the same subnet. With ARP takeover you can achieve availability that is to a high degree comparable to what can be achieved using dynamic routing.

Dependencies and restrictions of ARP takeover: ARP recovery solutions rely on the fact that hosts accept a new MAC address for a given IP on the same network medium. However, this has a major disadvantage in that other media such as HiperSockets or XCF cannot be used as alternate routes.

- ARP takeover requires dedicated OSA-Express adapters (ports) in QDIO mode.
- If you want to move an IP address from one adapter to another within a system you must also set up a Static Virtual IP Address (VIPA).
- While it is possible to configure and use multiple parallel paths to increase available bandwidth (called multi-pathing) without a dynamic routing protocol, if there is a failure in the network beyond a point where the OSA adapter is able to detect it, TCP connections that are directed across the failed path will time out and UDP and RAW traffic is lost.

For more information, basics and configuration samples on ARP high availability, refer to the redbook: *z/OS V1R13 Communications Server TCP/IP Implementation: Volume 3 High Availability, Scalability, and Performance*

z/OS TCP/IP configuration sample for the z/OS 1 LPAR (extract): The static VIPA of LPAR z/OS 1 is 10.101.6.188 and the SAP application talks to this VIPA only. The VIPA is mapped onto the real interface via the SOURCEVIPAINTERFACE statement, which makes the real interface IP transparent to the application. The z/OS 1 static VIPA 10.101.6.188 can be active either on the OSA interface with 10.101.6.140 or 10.101.6.141. The ARP message sent out is the one of the real interface.

```

DEVICE VLINK2   VIRTUAL 0           ; VIPA
LINK   VLINK2   VIRTUAL 0 VLINK2   ; VIPA
;
INTERFACE SYS1IF1                               ; INTERFACE TO SWITCH 1
DEFINE IPAQENET
IPADDR 10.101.6.140/26
PORTNAME SYS1P1
SOURCEVIPAINTERFACE VLINK2
MTU 8992
NONROUTER
;
INTERFACE SYS1IF2                               ; INTERFACE TO SWITCH 2
DEFINE IPAQENET
IPADDR 10.101.6.141/26
PORTNAME SYS1E2
SOURCEVIPAINTERFACE VLINK2
MTU 8992
NONROUTER
;
HOME
  10.101.6.188           VLINK2   ; VIPA
;
BEGINROUTES
; NETWORK      MASK  FIRST HOP   LINK      PCKTSZ
  ROUTE 10.101.6.128/26 =        SYS1IF1  MTU 8992
  ROUTE 10.101.6.128/26 =        SYS1IF2  MTU 8992
ENDROUTES
;
START SYS1IF1                               ; INTERFACE TO SWITCH1

```

```

START SYS1IF2                ; INTERFACE TO SWITCH2
;
BSDROUTINGPARMS TRUE
VLINK2 8992 0 255.255.255.240 0

```

Network switch

A single network switch would be a single point of failure (SPOF). Therefore a highly available network needs at least two switches with built-in redundancy features. In the sample configuration two CISCO 3750E switches are used. A Link Aggregation group is defined, covering the two switches, which are connected to each other to simulate a single virtual switch. Cisco calls this feature *Cross Stack EtherChannel*. It supports IEEE 802.3ad and LACP.

AIX network adapter

Our AIX environment has redundancy on the VIO Server level and within each VIO itself. There are two VIO servers per physical machine, each with two physical adapters. The adapters are configured as a Shared Ethernet Adapter (SEA) failover setup to achieve physical network adapter redundancy. The secondary adapter takes over when the primary adapter fails.

Advantages and disadvantages

Dedicated high availability tests have shown that this configuration can be an alternative setup to the recommended network setup with OSPF and multiple subnets. Usage of the simplified configuration depends on the customer requirements for network high availability and on the risk the customer can take.

Advantages:

- Simplified network setup with reduced complexity
- No dynamic routing configuration required
- One flat network
- Alternate routes can include HiperSockets. Starting with zEnterprise 196 GA2 processors, the Internal Queued Direct I/O (IQDIO) support offers another option besides the basic HiperSockets function, which can be used for alternate routes: the integration with the intraensemble data network (IEDN) controlled by the zManager functions which provide access controls, virtualization and management functions necessary to secure and manage the IEDN. This functionality is called extended IQD (IQDX). IQDX interfaces can now be dynamically detected and transparently merged with the IEDN. Then network traffic between systems on the same CEC on a z196 GA2 could use HiperSockets IQDX and IEDN paths transparently and would use external OSAx paths when necessary.

Disadvantages:

- Redundancy is only possible within the flat network. Other media such as HiperSockets or XCF cannot be used as alternate routes before zEnterprise 196 GA2.
- Redundancy is only given in failure cases, which are detectable through hardware. Redundancy is therefore not achieved, for example, if the OSA Express feature does not detect a loss of signal, TCP/IP has no knowledge of the failure of the network path. In that case it is up to TCP/IP time-outs to detect a failure in the network path and to terminate existing connections. A timeout is not a dedicated and meaningful error to TCP/IP so after a timeout there would be no trigger for TCP/IP to perform the takeover.

z/OS VIPA usage for the high availability solution for SAP

Certain static and dynamic VIPA definitions are required for the SAP HA solution.

For the SAP HA solution implemented with SA z/OS, it is necessary to create:

- Static virtual IP address (VIPA) definitions for:
 - z/OS systems that are hosting DB2 data sharing members
- Dynamic VIPA definitions for:
 - SCS and its corresponding ERS
 - disruptive VIPA for SCS
 - NFS server
 - SAP network interface router (SAProuter)

Note: Dynamic VIPAs are not supported by the standard *DB2 add-on policy.

- Dynamic non-disruptive VIPA definitions for:
 - RMF SAP Sysplex Monitoring

The dynamic VIPA is to be defined as VIPARANGE with the attributes MOVEABLE and DISRUPTIVE. The non-disruptive VIPA is to be defined with the attribute NONDISRUPTIVE:

```
VIPADYNAMIC
VIPARANGE DEFINE MOVEABLE DISRUPTIVE 255.255.255.240 10.101.5.192
VIPARANGE DEFINE MOVEABLE NONDISRUPTIVE 255.255.255.248 9.152.20.176
ENDVIPADYNAMIC
```

Furthermore, the SOURCEVIPA attribute is needed for all VIPAs.

Normally, with the SAP application server that is running on a non-z/OS system, all network session connects are inbound to z/OS. z/OS accepts the received source IP for the return of all IP packets.

However, when the Network Lock Manager (NLM) is used for the z/OS NFS Server then lock notification status causes z/OS to start the network session. It is recommended that you use a Dynamic VIPA for the NFS Server (because it can move across LPARs). Therefore, you must set a special source VIPA just for the NFS Server, which is supporting protocol version 3 clients. You set the source VIPA in the z/OS TCP PROFILE(s) by using the SRCIP statement as shown in the example.

```
SRCIP JOBNAME MVSNFSHA 10.101.5.193 ENDSRCIP
```

In this example, MVSNFSHA is the z/OS job name of the NFS server, and 10.101.5.193 is the dynamic VIPA associated with the NFS server.

The following PROCLIB member allows setting a dynamic VIPA by an operator command. System Automation can also call this procedure, substituting the IP address for the variable &VIPA. .

```
//TCPVIPA PROC VIPA='0.0.0.0'
//VIPA00 EXEC PGM=MODDVIPA,
// PARM='POSIX(ON) ALL31(ON)/-p TCPIP -c &VIPA.'
```

Note: The mentioned procedure is included as member INGEVIPA in the SINGSAMP sample library of System Automation for z/OS since its release 3.3. The SAP add-on policies that are included with System Automation for z/OS use

this procedure in the definition of the SAP VIPA resources. To enable the policy to call the procedure, you must copy it to one of your PROCLIBs.

If you use a TCPIP stack with a different name, then you need to change *TCPIP* to the name of your stack in your copy of the INGEVIPA procedure.

Considerations when running the AS to DB and SCS connections over a private subnet

If you want to use logon groups for SAP GUI users and do not want to expose IP addresses from your private network, which also covers the ASCS/SCS dynamic VIPA under z/OS UNIX (or AIX/Linux) to the public network used by SAP users, it is recommended that you use the SAP Router or SAP Web Dispatcher utilities or both.

For more information about these utilities go to the following links:

- ==> for the SAProuter
- ==> for the SAP WebDispatcher

Timeout behavior of the client/server connection over TCP/IP

In various subtopics, this topic handles the timeout behavior of a client/server connection with the TCP/IP communication protocol for each of the platforms AIX, Linux on z Systems, and Windows.

In conclusion, platform-independent information is then presented on the maximum transaction time. Because the timeout settings are system-wide settings, the timeout behavior applies, for example, to connections via DB2 Connect.

In order to optimize the availability of the SAP system it is essential that network failures are detected as early as possible. Therefore it is recommended that you change the default TCP/IP timeouts as described with this information.

Note: If you plan to change the default value of a timeout, please make sure that all DDF instances belonging to the same SAP system and all their corresponding clients use a similar value for that specific timeout.

Timeout behavior of the AIX application server

Various types of timeout related to the AIX application server are discussed in this topic.

On AIX, you can display and change your current network attribute values using the **no** command.

To avoid negative effects on system performance, it is recommended that you change default values only after careful study.

For more information on how the network attributes interact with each other, refer to *AIX System Management Guide: Communications and Networks*.

Client connection timeout

When the client connects to the server, each connection attempt times out after a time period determined by the value of the *tcp_keepinit* network attribute. When

this happens, the connect attempt has failed. The client then writes an error message and returns the error to the calling process.

The default value for *tcp_keepinit* is 75 seconds. This means that an AIX connection request times out after 75 seconds.

For an example of where you might need to change this default see “Additional considerations” on page 53. This setting is system-wide and affects all TCP connections established from the machine.

Related information:

- As you can read in “DB2 connection failover for ABAP instances” on page 16, the new SAP approach for implementing DB2 connection failover for ABAP instances is based on the native CLI driver failover capabilities.
- For the relationship between the setting of the TCP/IP client connection timeout and the DB2 connection timeout, see “Timeout behavior of DB2 client connections” on page 64.

Client transmission timeout

Each time the client sends data to the database server, TCP/IP waits for acknowledgment of this data. TCP/IP retransmits data if acknowledgments are missing. The time period that TCP/IP waits for the acknowledgment before it times out is variable and dynamically calculated. This calculation uses, among other factors, the measured roundtrip time on the connection. The timeout interval is doubled with each successive retransmission. When the final transmission timeout occurs, the client's next receive call fails with a send timeout error. The client writes an error message and returns the error to the calling process.

On AIX, the number of retransmissions is determined by the value of the *rto_length* network attribute.

The length of a transmission timeout on AIX is about 9 minutes, and is based on the default values of the following network attributes:

- *rto_length*, default is 13
- *rto_limit*, default is 7
- *rto_low*, default is 1
- *rto_high*, default is 64

which are used in calculating factors and the maximum retransmits allowable.

The following example shows how the AIX algorithm works:

There are *rto_length*=13 retransmission intervals. The first retransmission starts earliest after *rto_low*=1 second. The time between retransmissions is doubled each time (called exponential backoff). There are two parameters limiting the retransmission interval:

- *rto_limit*=7, which is the maximum number of such doubling and
- *rto_high*=64 seconds, which is the maximum interval between retransmissions.

For example, if you start with 1.5 seconds for the first retransmission interval, this leads to the following retransmission attempt times:

Table 3. Retransmission intervals

Transmission	Retransmission after (seconds)
1	1.5
2	3
3	6
4	12
5	24
6	48
7	64
8	64
9	64
10	64
11	64
12	64
13	(Reset)

After the 13th transmission attempt, TCP/IP gives up resending and sends a reset request.

Recommended values: For the client transmission timeout, it is recommended that you change the value of **rto_length** to **8**. This reduces the timeout to approximately 4 minutes.

Client idle timeout

If there is no data flow on a client/server connection, TCP/IP uses a so-called keep-alive mechanism to verify that such an idle connection is still intact after a predefined period of time. The term *idle* means with respect to TCP/IP and includes the case where the client is waiting in the **recv()** function because this waiting for data is a passive task and does not initiate any packet transfers for itself. If the remote system is still reachable and functioning, it acknowledges the keep-alive transmission.

On AIX, this mechanism is controlled by the network attributes **tcp_keepidle** and **tcp_keepintvl**.

The default values of these network attributes determine that an idle connection is closed after 2 hours, 12 minutes, and 30 seconds if no keep-alive probes are acknowledged.

Recommended values: It is recommended that these network attributes be set as follows:

- **tcp_keepidle** to **600** half-seconds (5 minutes) and
- **tcp_keepintvl** to **12** half-seconds (6 seconds).

This results in approximately 5 minutes + (10 * 6) seconds = 6 minutes.

Timeout behavior of the Linux application server

Various types of timeout related to the Linux application server are discussed in this topic.

On Linux on z Systems, you can display your current network attribute values by viewing the contents of the corresponding files in the directory `/proc/sys/net/ipv4`. Changing the file contents changes the parameter values.

To avoid negative effects on system performance, it is recommended that you change the default values only after careful study. A description of the different options can be found under the Linux Source Tree in the file `linux/Documentation/networking/ip-sysctl.txt`.

Client connection timeout

When the client connects to the server, each connection attempt times out after a time period determined by the value of the network attribute `tcp_syn_retries`. When this happens, the connect attempt has failed. The client then writes an error message and returns the error to the calling process. The default value for `tcp_syn_retries` is 5, which corresponds to about 180 seconds. This means that an Linux on z Systems connection request times out after 180 seconds. For an example where you might need to change this default, see “Additional considerations” on page 53.

This setting is system-wide and affects all TCP connections established from the system.

For the relationship between the setting of the TCP/IP client connection timeout and the DB2 connection timeout, see “Timeout behavior of DB2 client connections” on page 64.

Client transmission timeout

Each time the client sends data to the server, TCP/IP waits for acknowledgment of this data. TCP/IP retransmits data if acknowledgments are missing. The time period that TCP/IP waits for the acknowledgment before it times out is variable and dynamically calculated. This calculation uses, among other factors, the roundtrip time measured on the connection. The timeout interval is doubled with each successive retransmission (called *exponential backoff*). When the final transmission timeout occurs, the client's next receive call fails with a send timeout error. The client writes an error message and returns the error to the calling process.

On Linux on z Systems, the number of retransmissions is determined by the value of the network attribute `tcp_retries2`. The default value is 15, which corresponds to about 13-30 minutes depending on RTO.

Recommended values: For the client transmission timeout, it is recommended that you change the value of `tcp_retries2` to 8. This reduces the timeout to approximately 4 minutes.

Client idle timeout

If there is no data flow on a client/server connection, TCP/IP uses a so-called keep-alive mechanism to verify that such an idle connection is still intact after a

predefined period of time. The term *idle* means with respect to TCP/IP, and includes the case where the client is waiting in the **recv()** function because this wait for data is a passive task and does not initiate any packet transfers for itself. If the remote system is still reachable and functioning, it will acknowledge the keep-alive transmission. On Linux on z Systems, this mechanism is controlled by the network attributes **tcp_keepalive_time** (default is 2 hours), **tcp_keepalive_probes** (default value is 9) and **tcp_keepalive_intvl** (default value is 75 seconds). The default values of these network attributes determine that an idle connection is closed after about 2 hours and 11 minutes if no keep-alive probes are acknowledged.

Recommended values: It is recommended that these network attributes be set as follows:

- *tcp_keepalive_time* to 600 half-seconds (5 minutes)
- *tcp_keepalive_intvl* to 6 seconds.

This results in approximately 5 minutes + (9 * 6) seconds = 5 minutes and 54 seconds.

Timeout behavior of the Windows application server

Various types of timeout related to the Windows application server are discussed in this topic.

On Windows, the TCP/IP protocol suite implementation reads all of its configuration data from the registry. All of the TCP/IP on Windows parameters are registry values located under one of two different subkeys of `\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters`. Adapter-specific values are listed under sub-keys for each adapter identified by the adapter's globally unique identifier (GUID). The parameters described in this topic normally do not exist in the registry. They may be created to modify the default behavior of the TCP/IP protocol driver.

To avoid negative effects on system performance, it is recommended that you change default values only after careful study.

For more information about registry values, refer to the Windows online documentation and its references to TCP/IP documentation.

Client connection timeout

When the client connects to the server, each connection attempt times out after a time period determined by the value of the **TcpMaxConnectRetransmissions** registry value (under **Tcpip\Parameters**). When this happens, the connect attempt has failed. The client then writes an error message and returns the error to the calling process.

The default value of **TcpMaxConnectRetransmissions** is 3. The retransmission timeout is doubled with each successive retransmission in a given connect attempt. The initial timeout value is three seconds. This means that a Windows connection request times out after approximately 45 seconds.

For an example of where you might need to change this default see "Additional considerations" on page 53. This setting is system-wide and affects all TCP connections established from the system.

For the relationship between the setting of the TCP/IP client connection timeout and the DB2 connection timeout, see “Timeout behavior of DB2 client connections.”

Client transmission timeout

Each time the client sends data to the server, TCP/IP waits for acknowledgment of this data. TCP/IP retransmits data if acknowledgments are missing. The time period that TCP/IP waits for the acknowledgment before it times out is variable and dynamically calculated. This calculation uses, among other factors, the measured round-trip time on the connection. The timeout interval is doubled with each successive retransmission. When the final transmission timeout occurs, the client's next receive call fails with a send timeout error. The client writes an error message and returns the error to the calling process.

The length of a transmission timeout on Windows is determined by the **TcpMaxDataRetransmissions** registry value (under **Tcpip\Parameters**), and can amount to several minutes. The actual time is based upon the default value of **TcpMaxDataRetransmissions**, which is 5, and upon the initial timeout value, which depends on the measured roundtrip time on the connection as already mentioned. For example, if your initial timeout value is 2 seconds, then the transmission timeout is 2 minutes and 6 seconds.

Recommended values: It is recommended that you run with the default value.

Client idle timeout

If there is no data flow on a client/server connection, TCP/IP uses a so-called *keepalive* mechanism to verify that such an *idle* connection is still intact after a predefined period of time. The term *idle* means with respect to TCP/IP, and includes the case where the client is waiting in the **recv()** function because this waiting for data is a passive task and does not initiate any packet transfers for itself. If the remote system is still reachable and functioning, it will acknowledge the keep-alive transmission.

On Windows, this mechanism is controlled by the **KeepAliveInterval** and **KeepAliveTime** registry values (under **Tcpip\Parameters**).

The default values of these registry values determine that an idle connection is closed after two hours and six seconds if no keep-alive probes are acknowledged.

Recommended values:

It is recommended that you change the registry values of **KeepAliveInterval** to **360000** milliseconds (6 minutes).

This results in approximately 6 minutes + (6 * 1) seconds = 6 minutes and 6 seconds.

Timeout behavior of DB2 client connections

SAP application servers connect to the DB2 database on z/OS using the DB2 CLI driver (for ABAP instances) or the DB2 JDBC driver (for Java instances). A database connection attempt by an SAP work process might fail for a number of reasons, for example when the primary DB2 data sharing member is not available, or when the network connection from the application server to a z/OS LPAR is unavailable.

Using DB2 connection failover (see “DB2 connection failover” on page 15), SAP work processes try to connect to other DB2 data sharing members if their primary member is not available or not reachable. The time that an SAP work process waits connecting to a DB2 member and the maximum total time that an SAP work process waits when attempting to connect to all configured DB2 data sharing members depends on the following circumstances:

- the use of SAP failover (`connect.ini`) or the CLI failover (`db2dsdriver.cfg`) and parameter values that are specified in these files
- the profile parameters of the SAP application server instance and the settings of environment variables on the application server
- operating system settings like the client connection timeout (see the previous sections with the platform-dependent descriptions).

You might want to limit the combined maximum failover time for all connection attempts from an SAP application server to the database. To do so you should consult the settings for the failover option (SAP failover or CLI failover) that you are using.

For details on how to limit the single and the maximum total DB connection times, refer to:

- *SAP Note 1465252: DB2 z/OS: Exploit CLI time out parameter* for the SAP failover option
- the *Database Administration Guide: SAP on IBM DB2 for z/OS* for the CLI failover option.

When converting from SAP failover to CLI failover, you need to carefully configure the parameters to achieve the same timeout behavior - for example a maximum timeout of 20 seconds per database connection attempt - as before.

Note: Ensure that you do not set any parameters that cause the connection failover mechanism to stop before connections to all configured failover DB2 data sharing members have been attempted.

Timeout behavior of the database server

Various types of timeout related to the database server are discussed in this topic.

The following definitions of timeout values pertain to the standard TCP/IP communication protocol.

On z/OS, you can check your current TCP/IP parameter values by looking at the PROFILE.TCPIP data set. For details on the PROFILE.TCPIP statements, refer to *z/OS Communications Server: IP Configuration Reference*.

Server transmission timeout

Each time a server thread sends data to the client, TCP/IP waits for this data to be acknowledged. If acknowledgments are missing, TCP/IP retransmits data. The time period that TCP/IP waits for the acknowledgment before it times out is variable and is calculated dynamically. This calculation uses, among other factors, the measured round-trip time on the connection.

The number of retransmissions is determined by the following values, which are parameters of the GATEWAY statement in the PROFILE.TCPIP data set:

- *MAXIMUMRETRANSMITTIME*, default is 120 seconds

- *MINIMUMRETRANSMITTIME*, default is 0.5 seconds
- *ROUNDTRIPGAIN*, default is 0.125
- *VARIANCEGAIN*, default is 0.25
- *VARIANCEMULTIPLIER*, default is 2.00

It is recommended to use the default values unless you find your retransmission rate is too high. When the final transmission timeout occurs, the server thread's next receive call fails with a send timeout error. The server thread writes an error message and exits.

Server idle timeout

If there is no data flow on a client/server connection, TCP/IP uses a so-called *keep alive* to verify that such an idle connection is still intact after a predefined period of time. The keep-alive mechanism sends keep-alive probes to the other end. If the partner system is still reachable and functioning, it will acknowledge one keep-alive probe and TCP/IP will wait again until it is time for another check. If several keep-alive probes are not acknowledged, TCP/IP deems the connection broken and gives control to the server thread, which in turn writes an error message and exits.

The system-wide value defining the time after which a TCP/IP connection with no data flow is verified is set in the `KEEPALIVEOPTIONS` statement in the `PROFILE.TCPIP` data set. In the following example, a value of 10 is used, meaning that the first keep-alive probe is sent after ten minutes:

```
KEEPALIVEOPTIONS
INTERVAL 10
ENDKEEPALIVEOPTIONS
```

If such a statement is not contained in the `PROFILE.TCPIP` data set, the default time is 2 hours. After that time, the TCP/IP keep-alive mechanism sends up to ten keep-alive probes in intervals of 75 seconds. If no probe is acknowledged, this translates into 12 minutes and 30 seconds. Together with the default time of 2 hours, this means that an "idle" connection is regarded as broken after 2 hour, 12 minutes, and 30 seconds. Note that with the minimum value of 1 minute for the `INTERVAL` option, this time is still 13 minutes and 30 seconds.

DDF-specific keep-alive interval times

The default value for the TCP/IP keep-alive interval with DDF is 120 seconds (`(DSNZPARM: DSN6FAC TCPKPALV)`). This value is less than the default value for DB deadlock and timeout detection (which is normally 10 minutes) and guarantees that a DDF thread with a broken connection will not hold a DB2 resource long enough that another DDF thread encounters a DB2 resource or deadlock timeout. It is recommended that you run it with the default value.

Resource timeout and deadlock detection interval

The following DB2 subsystem parameters control the resource timeout and deadlock detection interval.

Resource timeout

The parameter `DSNZPARM: DSN6SPRM IRLMRWT` (recommended value: **600**) specifies the length of time (in seconds) the Internal Resource Lock Manager

(IRLM) waits before detecting a timeout. The term *timeout* means that a lock request has waited for a resource longer than the number of seconds specified for this parameter. The value specified for this parameter must be an integer multiple of the DEADLOCK TIME because IRLM uses its deadlock timer to initiate both timeout detection and deadlock detection.

Deadlock detection interval

The parameter *IRLM PROC: DEADLOCK* (first value; recommended value: 5) specifies the length of time (in seconds) of the local deadlock detection cycle. A deadlock is a situation where two or more DB2 threads are waiting for resources held by one of the others. Deadlock detection is the procedure by which a deadlock and its participants are identified.

The deadlock detection cycle should be shorter than the resource timeout.

The maximum time to detect a deadlock is two times the deadlock detection cycle.

SAP maximum transaction time

SAP has a concept of limiting the transaction time to a maximum. Each transaction's maximum time depends on the value of the SAP instance profile parameter *rdisp/max_wprun_time* (in seconds). The default value of *rdisp/max_wprun_time* is 600.

The total time until the short dump is issued is called *total maximum transaction time*. The formula to calculate the total maximum transaction time is:

$rdisp/max_wprun_time + 60$

seconds.

The default time is thus:

$\underline{600} + 60 = \underline{660}$

seconds.

When this time elapses, an ABAP short dump is issued.

Chapter 5. Concepts for a high availability SAP solution

Read this information to learn about prerequisites and planning considerations for business continuity for SAP on IBM z Systems. The contained subtopics also describe the architecture of the high availability solution for SAP and its system infrastructure requirements.

The information is structured into the following subtopics:

- “Prerequisites and planning”
- “Architecture components”
- “Failure scenarios and impact” on page 83

Prerequisites and planning

In a standard distributed SAP environment, the database service (server), the SAP Central Services and the NFS service (server) are the so-called *single points of failures* (SPoFs). In order to minimize the impact of the outage of one of the SPoF services, it is necessary to set up redundancy. This means running one or more (standby) servers where each of the SPoF services can be failed over and restarted independently.

- Redundancy for the database is achieved by using DB2 Data Sharing, a true parallel data base setup. Use a static z/OS virtual IP address (VIPA) for each LPAR running one of the data sharing members for network high availability reasons.
- If SAP Central Services or NFS server must be moved, it is essential to allow the rest of the SAP components to reestablish their connections to the moved SPoF service after such a failover or restart. For this situation, each SPoF service must have its own associated virtual host name.
- Moving a service together with its VIPA allows the previously connected SAP components to find the moved or restarted service again by attempting to reconnect to the same virtual host name.

SAP high availability installation

Based on the prerequisites, you MUST install the DB2 database in data sharing mode. Install the ABAP Central Services and the Java Central Services and their corresponding replication server instances on z/OS with the REXX-based ZSCSinst installation tool. For each instance, use its own virtual host name. Use the SAPinst tool with parameter SAPINST_USE_HOSTNAME specifying the virtual host names for installations on AIX or Linux.

Architecture components

This information discusses the involved architecture components for a high availability solution for SAP.

- SAP Central Services (SCS)
- Fault tolerant network
- File system considerations
- Database considerations
- Applications designed for a highly available environment

SAP Central Services

The SAP Central Services concept is applicable to both ABAP and Java systems. By implementing high availability for SAP Central services you can ensure that vital services in an SAP system (enqueue and message service) are accessible at all times.

Note: SAP has designated SAP Central Services for ABAP as *ASCS* (ABAP SAP Central Services) and now applies the abbreviation *SCS* to the Java-based variant. This is attributable to the use of these abbreviations as directory names. However, this publication continues to use the abbreviation *SCS* as a conceptual term and to refer to an *SCS* instance in general terms. It employs *ASCS* and *Java SCS* to distinguish the environment-dependent instances.

Earlier SAP releases were based solely on the central instance concept. This concept provided the following functionality:

- It hosted the enqueue work process
- It usually served as location of the message server and the syslog collector
- It hosted an SAP gateway process and serves as primary destination for RFC connections

Generally, 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.

To remove the central instances as a single point of failure and thereby enable high availability solutions, the central instance has been disassembled and redesigned into standalone components that operate as SAP Central Services (SCS). The current SAP releases now offer an HA option in their installation procedure, which installs the *SCS* instead of the traditional central instance.

If you have an SAP system with the old central instance (CI) concept that you want to enable for high availability, then you must split the CI into a Central Services (*ASCS*) instance and a remaining primary application server (*PAS*). For an overview and rationale for this procedure, refer to this article: *Why splitting off the ASCS from PAS?*

The following SAP notes contain instructions how to split the central services using SAP tool support:

- **SAP Note 2073500: FAQ: Splitting off ASCS from PAS**
- **SAP Note 2119669: How to split the ASCS from Primary Application Server (PAS)**

With *SCS* the independence of the components allows for a 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 as *SCS*:

- enqueue server
- message server
- SAP gateway server

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

Furthermore, the components of SCS 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.

Standalone enqueue server and enqueue replication 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.

Within the SAP Central Services, the enqueue server is a standalone component. The 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). See Figure 13 on page 72.

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 SCS. It is an own SAP instance, named ERS<xx> and must be installed with its own VIPA.

The multithreaded 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.

With SAP kernel 7.21 or higher, an alternative enqueue replication mechanism is available for SAP on z Systems. This mechanism does no longer require an enqueue replication server instance, the backup of replication data is written to the z/OS coupling facility instead. For details see Chapter 13, “Enqueue replication into a z Systems coupling facility,” on page 277.

Failover and recovery of SAP Central Services

Figure 13 on page 72 shows the principal TCP/IP communication paths between the application server instances and the enqueue and message servers. The VIPA used for ERS is not shown because it is of minor relevance for the failover scenario.

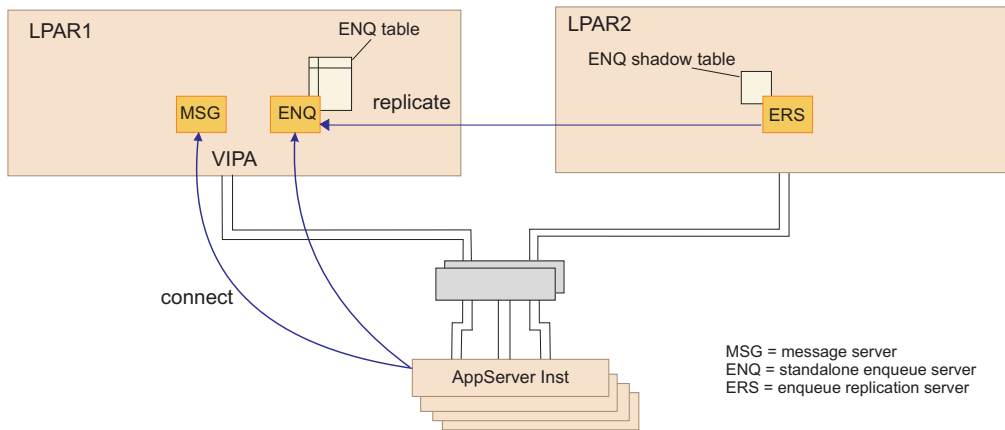


Figure 13. Initial startup of SCS

If the first system fails, the second system takes over the role of the first one, as shown in Figure 14:

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.

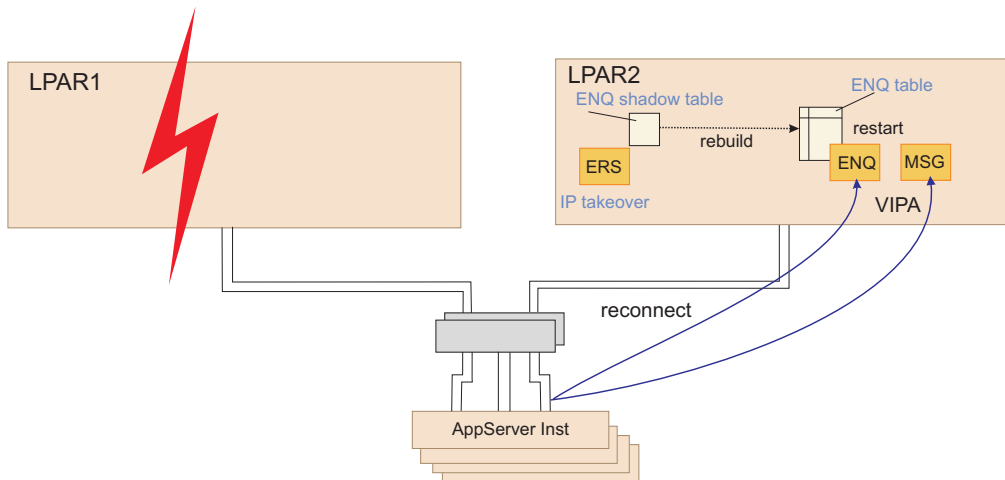


Figure 14. Failure of SCS and recovery of the enqueue table

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 15 on page 73.

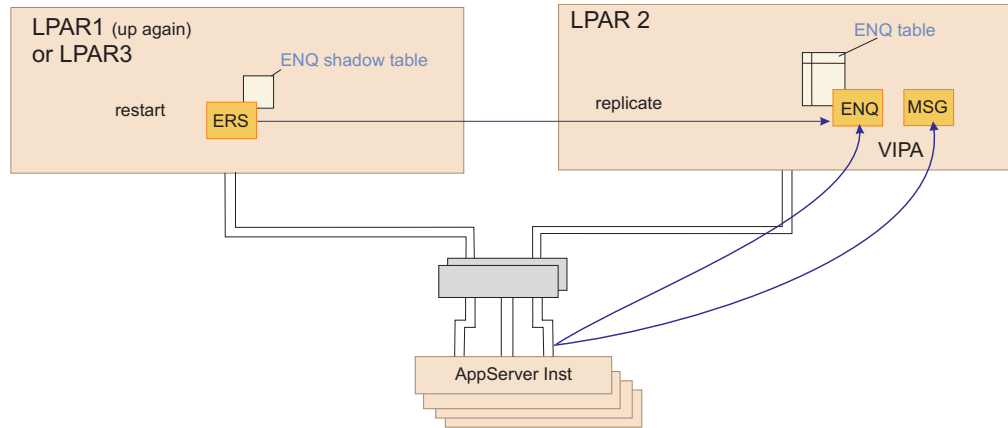


Figure 15. Movement of the enqueue replication server

Failover and recovery of SAP Central Services using EnqCF replication

This topic outlines the failover and recovery scenario when using the alternative enqueue replication mechanism that uses the z Systems coupling facility (also referred to as *EnqCF replication*).

Figure 16 shows the principal TCP/IP communication paths between the application server instances and the enqueue and message servers. It also shows the links that exist between each LPAR in the sysplex and the coupling facility.

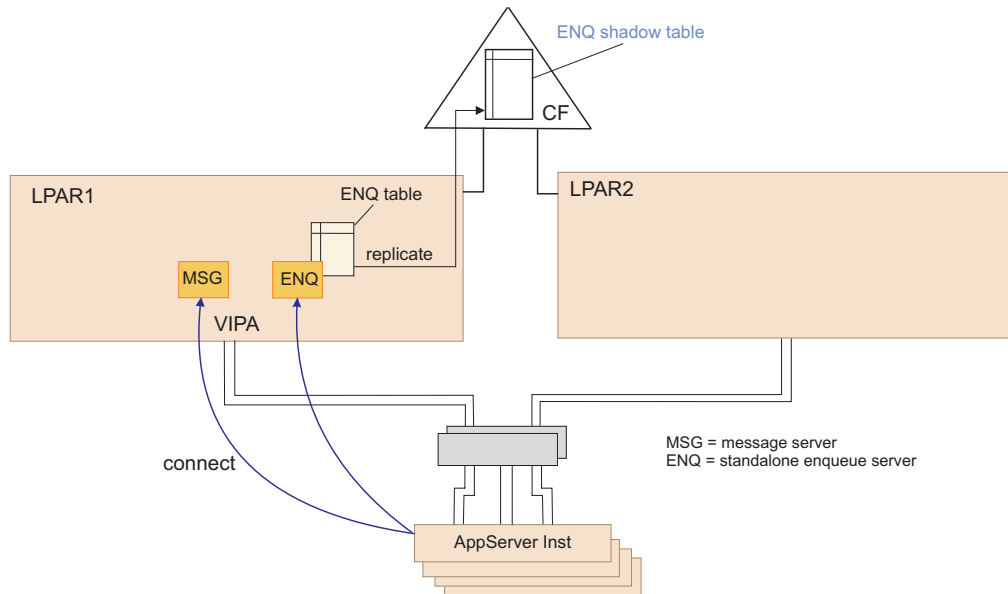


Figure 16. Startup of SCS during failover and recovery using EnqCF replication

If LPAR1 fails, LPAR2 takes over its role, as shown in Figure 17 on page 74.

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 in the coupling facility.
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.

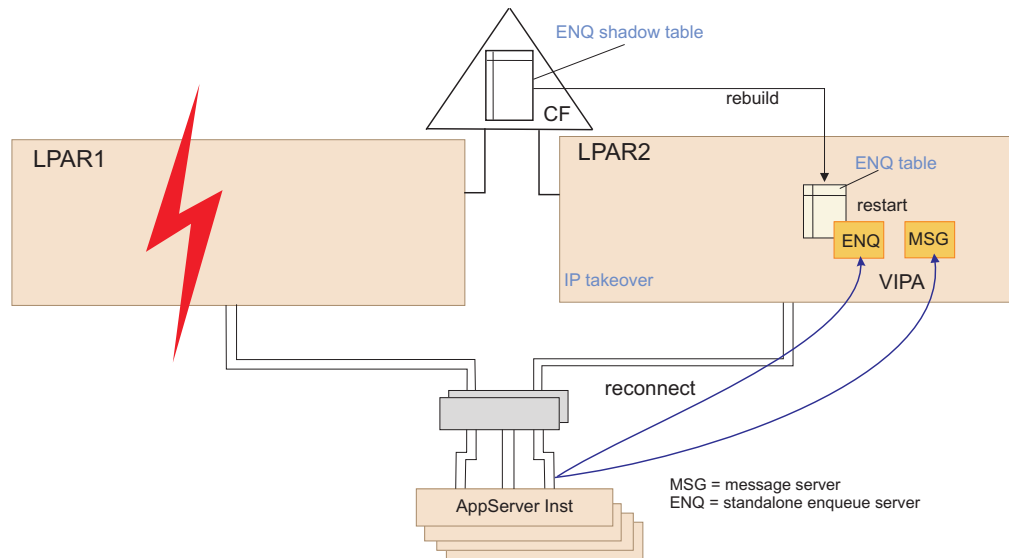


Figure 17. Failure of SCS and recovery of the enqueue table using EnqCF replication

Network

An SAP high availability solution requires a fault tolerant network. Therefore, to protect against network failures, all network components need to be duplicated.

IBM platforms (z/OS, Linux on z Systems, 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). A fault-tolerant network can be, for example, the intraensemble data network (IEDN) between IBM zEnterprise processors and an IBM zEnterprise BladeCenter Extension (zBX).

Static VIPAs are used to locate a host, while *dynamic VIPAs* are used to locate an application. Note that an application can be moved between hosts and can activate a dynamic VIPA on the host on which it is running.

For a fault-tolerant network, it is furthermore recommended to define a 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 18 on page 75 shows the general concept of a fault-tolerant network with duplicated network components and VIPA.

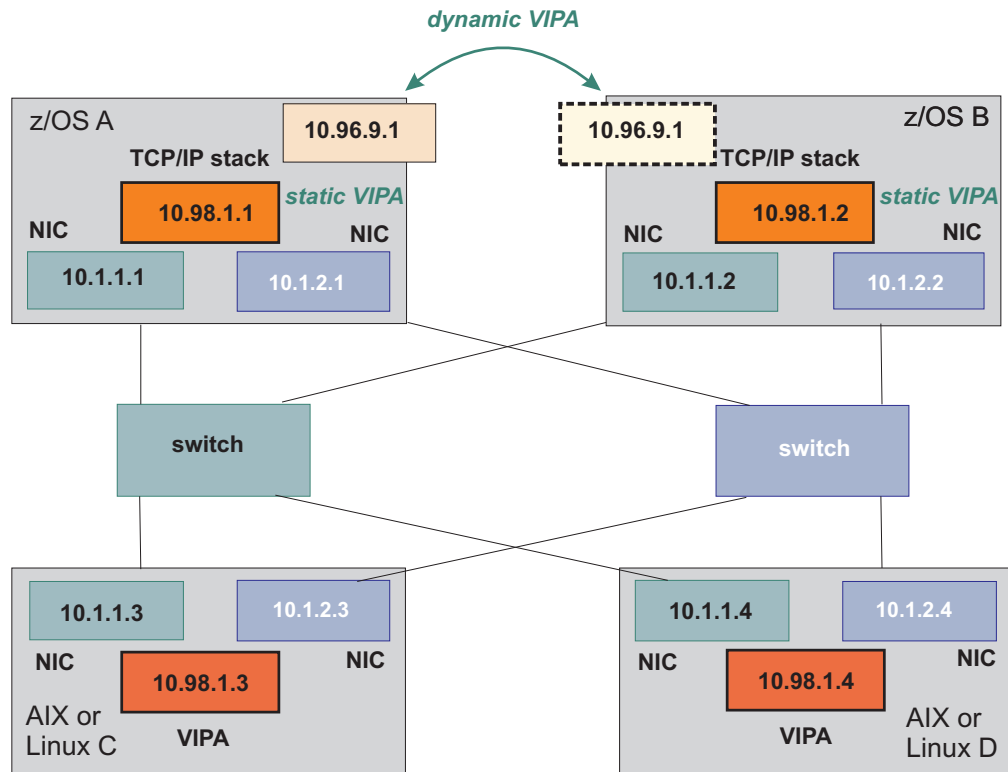


Figure 18. General concept of a fault-tolerant network with dynamic routing and four subnets

This fault-tolerant network concept is applicable to the connection between a remote SAP application server and the SCS as well as to that between a remote SAP application server and the DB2 on z/OS database server. See Chapter 4, "Network characteristics for high availability," on page 39 for details on how to set up a highly available network like shown in Figure 18, or for details about the alternative setup like the intraensemble data network (IEDN) of zEnterprise and zBX, which does not need dynamic routing.

The following figures show how dynamic rerouting works. In Figure 19 on page 76, the virtual IP address `virt_addr_1` on system A can be reached through IP addresses `addr_1` and `addr_2`. 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.

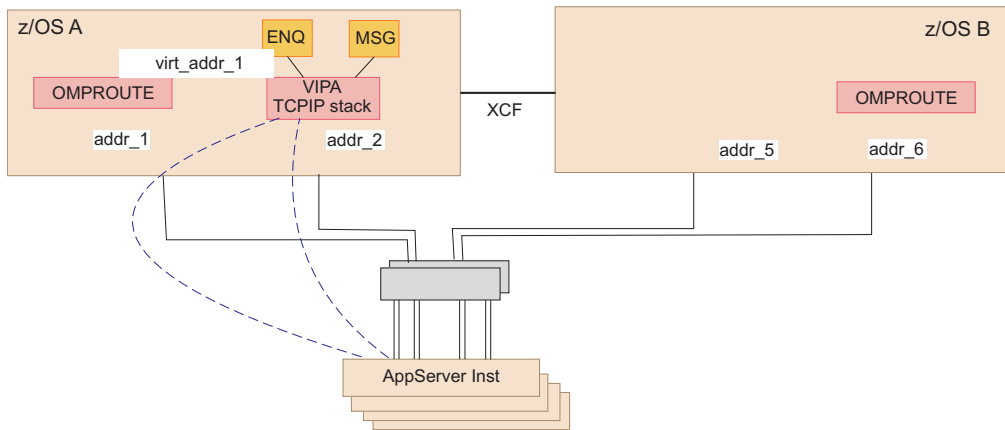


Figure 19. Alternative paths in a duplicated network

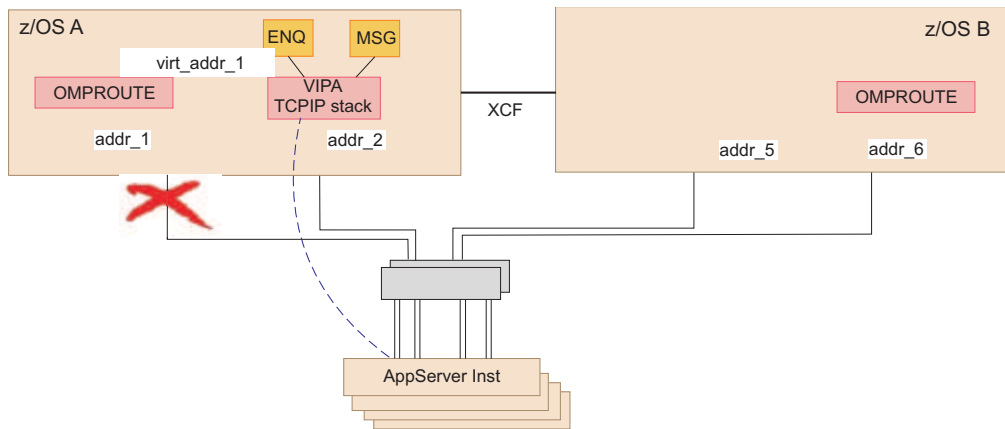


Figure 20. Rerouting if a network adapter card fails

What happens if network adapter card addr_1 fails? As shown in Figure 20 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 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 21 on page 77. 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.

Defining DISRUPTIVE VIPAs in z/OS ensures 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.

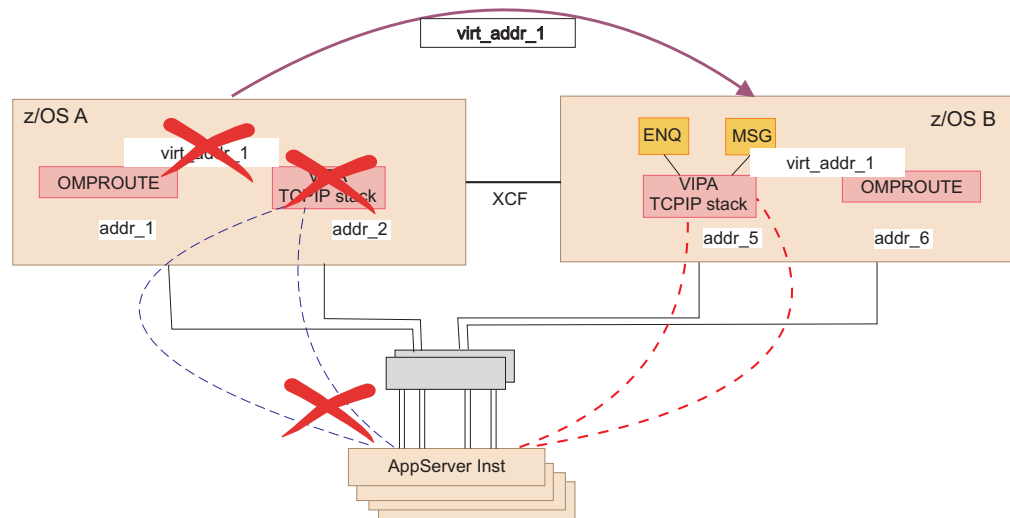


Figure 21. VIPA takeover and dynamic routing

In the scenario described in this book, 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. For a HiperSockets "only" 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. But in an HA environment, which comprises at least a second z Systems CEC, LAN connections to that second CEC are necessary for a DB or SAP Central Services failover. So in an HA environment always use VIPA also on Linux.

Static VIPAs are used for components *that are not moved* between systems, like DB2 DDF address spaces or SAP 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
- ABAP and Java SCS
- ERS of ABAP SCS
- ERS of Java SCS
- SAP network interface router (SAProuter)

While the rerouting shown in Figure 19 on page 76 through Figure 20 on page 76 is applicable to both static and dynamic VIPAs, the takeover shown in Figure 21 applies to dynamic VIPAs only.

As previously noted, the concept of a fault-tolerant network relates to the connection between

- remote SAP application servers and SCS
- remote SAP application servers and the DB2 on z/OS database server.

It is not necessary to introduce dynamic routing on SAP presentation servers (systems running the SAP GUI interface) in order to get to the message server via the VIPA of SCS. Such a connection to the message server is established at group logon time for example. You define a subnet route to the SCS VIPA via the normal

SAP application server subnet that the presentation server uses to access the SAP application server itself. When IP forwarding on the SAP application servers is enabled, OSPF will automatically route correctly from the SAP application server to the VIPA of the SCS and back.

File system

The SAP system requires shared access to some directories (global, profile, trans), while shared access is optional for other directories (for example, the directory containing the executable programs). We highly recommend to use NFS also for the global exe directory/directories. In addition, failover needs to be considered if connections to the directories is disrupted.

Shared directory access between z/OS systems is best achieved by using shared zFS file systems. The use of shared HFS is no longer recommended because it may become obsolete. New file systems should be created as zFS. For more information about how to migrate HFS to zFS file system, see

<http://www.redbooks.ibm.com/abstracts/TIPS0647.html>

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

For UNIX or Linux 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.

Note: Starting with Windows 2008, you should use Windows Services for NFS to access remote file systems. Windows does currently not support NFS version 4 (NFSv4), therefore you must use NFSv3. For details and setup of a SA z/OS NFS HA installation, and of an SAP application server on Windows, refer to the SAP SDN article: *SAP HA installations on z/OS with application servers on Windows*.

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. Therefore, with NFS version 3 (NFSv3), the Network Lock Manager (NLM) must be used to guarantee transactional file access. With NFS version 4, the locking function is automatically enabled. The methods described in the next section of this topic ensure that the NFS file systems become available again, quickly and automatically. In most cases this is transparent to the SAP system. See also “Application design” on page 83.

High Availability and performance improvements with zFS sysplex-aware

zFS provides additional performance improvements when running sysplex-aware in a shared file system environment. Also, in order to support read/write mounted file systems that are accessed as sysplex-aware, zFS automatically moves zFS ownership of a zFS file system to the system that has the most read/write activity.

It is recommended to define NFS exported zFS shared file systems as sysplex-aware. For details see *z/OS V2R1.0 Distributed File Service zFS Administration Guide*.

Performance improvements and space efficiency with z/OS file system aggregate version 1.5 and extended (v5) directory

Beginning with z/OS V2R1, zFS provides an optional, new format zFS aggregate, the version 1.5 aggregate. One purpose of the version 1.5 aggregate is to support a new directory format (extended v5 directory) that will scale better when the directory contains many names (over 10,000).

Extended (v5) directories provide the following benefits:

- They can support larger directories with performance.
- They store names more efficiently than v4 directories.
- When names are removed from extended (v5) directories, the space is reclaimed, when possible.

Earlier z/OS releases cannot access extended (v5) directories or version 1.5 aggregates. In order to control the transition to the new format directories, extended (v5) directories can only be created in version 1.5 aggregates.

Note: You should only create or change to a version 1.5 aggregate if you are sure you will not run z/OS releases prior to V2R1 in your sysplex. To create or change to a version 1.5 aggregate requires explicit action. By default, aggregates created in z/OS V2R1 are version 1.4 aggregates. Over time, it is likely that the default changes to version 1.5 aggregates.

For more information on v1.5 aggregates and v5 directories, refer to the *z/OS V2R1.0 Distributed File Service zFS Administration Guide*.

Migration prerequisite: All members of the sysplex must be z/OS V2R1 or higher.

Migration considerations:

- Identify the file system aggregates that you would like to convert to v1.5 aggregates. Display the actual version of an aggregate using the command:
zfsadm aggrinfo -aggregate omvs.zfs.sapmnt -long
- Identify the directories in those file systems that you would like to convert to v5 directories. Display the actual version of a directory using the command:
zfsadm fileinfo -path /sapmnt

Migration procedure with down time: This variation is only applicable if the file system can stay unmounted for a moment.

Note: In a shared FS configuration, the file system must stay unmounted on every system.

1. Unmount the file system using the command:

```
/usr/sbin/unmount /sapmnt
```

2. Edit the BPX PARMLIB member that holds the file system mount commands and insert **PARM('CONVERTT05')** like in this sample:

```
MOUNT FILESYSTEM('OMVS.ZFS.SAPMNT')  
MOUNTPOINT('/sapmnt')  
TYPE(ZFS)  
PARM('CONVERTT0V5')  
MODE(RDWR)  
AUTOMOVE
```

With the first mount the file system will be converted to v1.5 aggregate. Also all directories including sub-directories in the file system will be converted to v5 directories.

Migration procedure without down time: Unmount of file systems is not required.

1. Display actual version of an aggregate:

```
zfsadm aggrinfo -aggregate omvs.zfs.sapmnt -long
```

2. Convert this aggregate to v1.5:

```
zfsadm convert -aggrversion OMVS.ZFS.SAPMNT
```

Message IOEZ00810I indicates the successful change: Successfully changed aggregate OMVS.ZFS.SAPMNT to version 1.5.

3. Display actual version of a directory:

```
zfsadm fileinfo -path /sapmnt
```

4. Convert a single large directory to v5:

```
zfsadm convert -path /sapmnt
```

Message IOEZ00791I indicates the successful conversion: Successfully converted directory /sapmnt to version 5 format.

Note: There is no known option to include sub-directories. To convert a large number of directories you might consider creating a script. The approximate rate of conversion is hardware dependent and is 10.000 directory entries per second for a zEnterprise EC12 machine.

All newly created file system aggregates will be v1.5 aggregates (change the default):

1. Verify actual setting:

```
zfsadm configquery -format_aggrversion
```

2. Configure v1.5:

```
zfsadm config -format_aggrversion 5
```

3. Verify the new setting:

```
zfsadm configquery -format_aggrversion
```

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:

- A dynamic VIPA is defined that moves with the NFS server.
- The NFS clients must use the dynamic VIPA as host name in their mount command.
- The physical file systems that are exported by the NFS server must be accessible on all systems where the NFS server might be possibly started. This is another reason for using zFS shared file system sysplex-aware support.

The failover scenario is shown in Figure 22 on page 81 and Figure 23 on page 81. Note that the NFS VIPA is different from the VIPA of SCS. So they can be moved independently of each other.

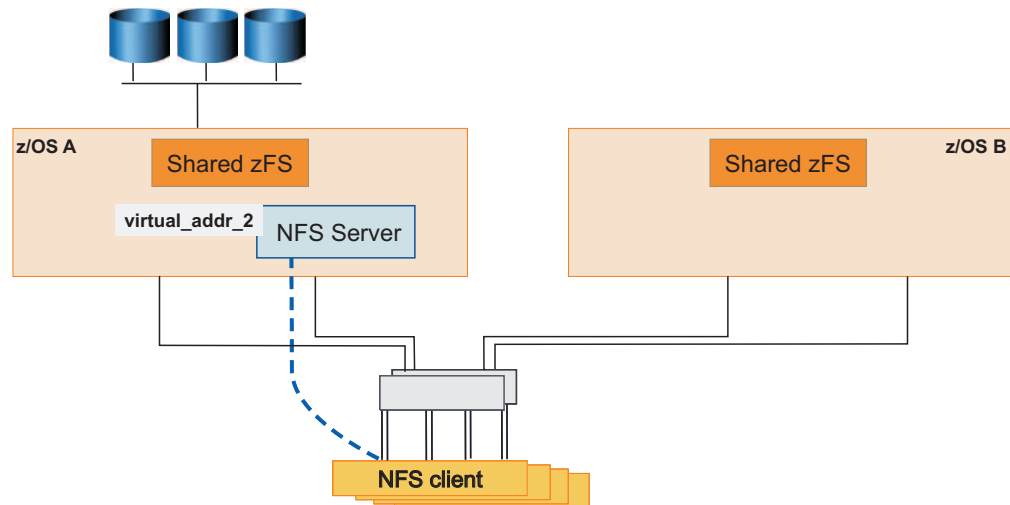


Figure 22. Initial NFS client/server configuration

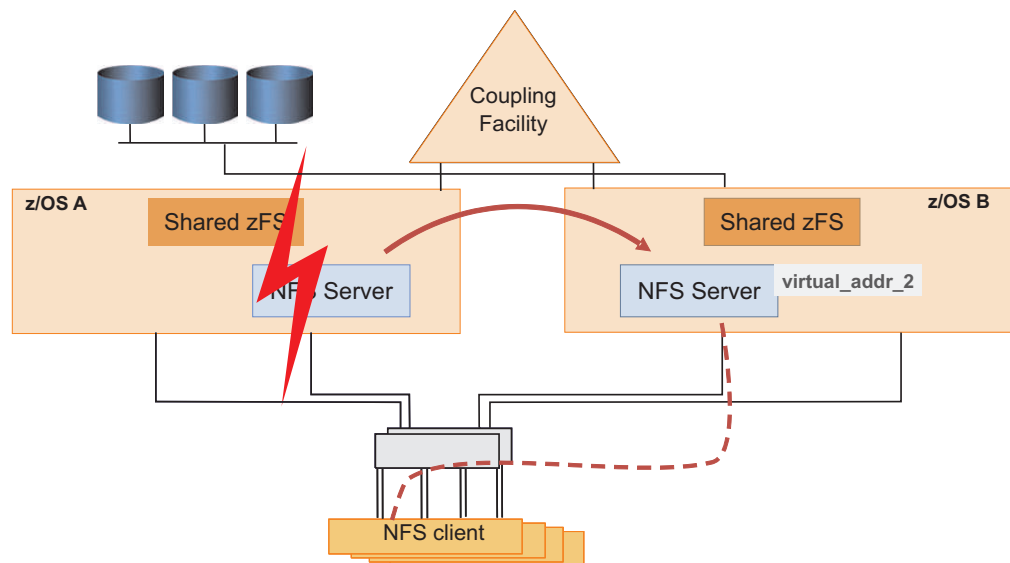


Figure 23. Failover of the NFS server

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 “SAP Central Services” on page 70.

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.

It is recommended that you use SA z/OS for automating the starting, stopping, monitoring, and restarting of the database server. With SA z/OS you are able to achieve 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.

Be aware that the DB2 utilities like RUNSTATS, REORG or BACKUP SYSTEM are designed to work in an online fashion. Therefore, the SAP and DB2 systems can remain up and running while you run utilities. Also, a large number of DB2 system parameters (ZPARMs) can be changed while the DB2 system remains up and running.

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, for example z/OS Automatic Restart Manager, that can be utilized to minimize their impact by quickly restarting the database server.

SA z/OS provides the added advantage of automating daily operational activities such as starting, stopping, and monitoring the entire SAP system, including the database server. SA z/OS also ensures that components are started and stopped in the proper sequence. The automating of these activities provides for quicker SAP system start-ups 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 DB2 connection failover (see "DB2 connection failover" on page 15) to move the 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, DB2 connection failover 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 previously stated, 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.

For a more detailed description of the SAP profile parameters that influence failover support, refer to the *Database Administration Guide: SAP on IBM DB2 for z/OS*, section *Sysplex Failover and Connection Profile*.

Note: In a data sharing environment, recovery from any database server and network failure can be achieved with the DB2 connection failover by switching over to a standby database server. But such a switch over is not transparent. To recover network failures, however, you must provide the appropriate redundancies, such as duplicate LAN connections and switches.

Application design

The hardware, operating system, database, middle ware, 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 inflight 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.

Failure scenarios and impact

This information discusses the impact that various failure scenarios have on the SAP system end user. For all the configurations discussed, it is assumed that SA z/OS is being used.

Without SA z/OS, the impact on the SAP system is very different to the description in the **Impact** column in Table 5 on page 85.

Without SA z/OS, all recovery actions must be done manually. Generally recovery takes longer and is error-prone when manual recovery procedures are performed under the pressure of a system outage. At best such manual recovery actions cause SAP transactions to time out and roll back.

When System Automation for Multiplatforms is running, it is possible to run the NFS server (file systems), SCS, application server instances, and SAProuter under control of Tivoli System Automation for Multiplatforms.

The scenarios outlined in Table 5 on page 85 are those that are of most concern to users. They are a subset of the scenarios discussed in “Verification procedures and failover scenarios” on page 193.

The scenarios are described for the following high availability configurations:

Table 4. High availability configurations

HA abbreviation used in Table 5 on page 85	Description	Notes
No HA	Old-style central instance with no data sharing implemented for the DB2 server.	You should use this configuration only for SAP systems for which high availability is not required. Caution: Database, central instance, and network are single points of failure. Failures of these critical components impact the whole SAP system.
Medium-scale HA	Data sharing, DB2 connection failover, double network (single central instance)	This scenario builds on the previous scenario by adding DB2 data sharing, DB2 connection failover, shared zFS filesystems, and a highly available network with VIPA and OSPF. This scenario still uses the old-style central instance. Note: Redundancy and failover capabilities are implemented for database and network. The central instance (inclusive message server) remains a single point of failure.
Full-scale HA	Enqueue replication and NFS failover, fully functional high availability	This scenario builds on the previous two scenarios by adding the SCS, the enqueue replication server (if applicable), and NFS failover support. This scenario is the fully implemented high availability solution for SAP. Note: 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. The implementation of this scenario is described in Chapter 6, “Preparing a high availability SAP solution,” on page 89.

In Table 5 on page 85, *SA* indicates actions taken automatically and instantaneously by SA z/OS or System Automation for Multiplatforms, and *User* indicates actions taken by the user.

To perform the action **User: Restart transactions**, you could consider using workload scheduling software, for example, Tivoli Workload Scheduler.

Table 5. Failure scenarios and impact

Failure	HA Configuration	Impact	Actions
DB2	No HA	<ul style="list-style-type: none"> Rollback of transactions Application servers wait until DB2 is up again 	SA: Restart DB2 User: Restart transactions
	Medium-scale HA and Full-scale HA	<ul style="list-style-type: none"> Rollback of transactions Remote application servers failover to other DB2 systems 	
Central instance	No HA and Medium-scale HA	<ul style="list-style-type: none"> Rollback of transactions Application servers wait until central instance is up again 	SA: Restart central instance User: Restart transactions
	Full-scale HA	N/A, see impact of enqueue server failure instead	N/A
Enqueue server	No HA and Medium-scale HA	N/A, see impact of central instance failure instead	N/A
	Full-scale HA	None	Using standard TCPIP-based replication: <ul style="list-style-type: none"> SA: Failover enqueue server to LPAR where enqueue replication server was running. SA: Move enqueue replication server Using replication into the CF: <ul style="list-style-type: none"> SA: Restart enqueue server on same or different LPAR
Enqueue replication server	No HA and Medium-scale HA	N/A	N/A
	Full-scale HA	None	SA: Restart enqueue replication server
Message server	No HA and Medium-scale HA	<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 User: Restart transactions
	Full-scale HA	<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

Table 5. Failure scenarios and impact (continued)

Failure	HA Configuration	Impact	Actions
Application server instance	No HA and Medium-scale HA and Full-scale HA	<ul style="list-style-type: none"> • Transactions on this instance are lost • Rollback of database updates • User sessions on this instance are lost 	User: connect to another instance User: Restart transactions SA: Restart instance
SAP gateway	No HA and Medium-scale HA and Full-scale HA	<ul style="list-style-type: none"> • For most transactions, no impact • Connections to registered RFC servers inhibited until they have reconnected to the SAP gateway 	SA: Restart SAP gateway
SAProuter	No HA and Medium-scale HA and Full-scale HA	<ul style="list-style-type: none"> • User sessions lost • Reconnect inhibited 	SA: Restart SAProuter User: Reconnect
NFS server	No HA and Full-scale HA	<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 User: Restart transactions
	Full-scale HA	<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	No HA	<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 	User: Recover and remount the file system User: Restart transactions
	Medium-scale HA and Full-scale HA	<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 	User: Restart transaction
Network (router, switch, adapter card)	No HA	<ul style="list-style-type: none"> • Lost connectivity to message server and SAP gateway server (see failures of these components) • Rollback of transactions on remote application servers • Remote application servers wait until network is up again 	User: Resolve network problem User: Restart transactions
	Medium-scale HA and Full-scale HA	None	None

Table 5. Failure scenarios and impact (continued)

Failure	HA Configuration	Impact	Actions
TCP/IP on SCS	No HA and	N/A	N/A
	Medium-scale HA		
	Full-scale HA	Enqueue server, message server and SAP gateway fail (see failures of individual components)	SA: Restart TCP/IP SA: Restart enqueue server, message server, SAP gateway
TCP/IP on database server	No HA and Medium-scale HA and Full-scale HA	Connection to database server lost (see failure of DB2)	SA: Restart TCP/IP User: Restart transactions
z/OS LPAR	No HA and Medium-scale HA and Full-scale HA	All components running in the LPAR fail (see failures of individual components)	User: Restart of LPAR SA: Restart DB2 SA: Restart other components

Chapter 6. Preparing a high availability SAP solution

This information unit describes planning tasks to prepare a new, or enable an existing SAP on DB2 for z/OS system for high availability using Tivoli System Automation for z/OS and Tivoli System Automation for Multiplatforms.

The contained information is structured into the following topics:

- “Software prerequisites” on page 90
- “Naming conventions” on page 91
- “Setting up DB2” on page 95
- “Setting up file systems under z/OS for local and NFS exported file systems” on page 95
- “Configuring Sysplex performance monitoring highly available with the RMF Distributed Data Server (RMF DDS)” on page 104
- “Modifying the environment for AIX and for Linux on z Systems” on page 106
- “Recommendations for sysplex failure management (SFM) policy definitions” on page 109
- “Setting up Tivoli System Automation” on page 109
- “SAP installation planning considerations” on page 109

Figure 24 on page 90 shows a sample system configuration where all SAP components can be made highly available by using SA z/OS and optionally System Automation for Multiplatforms (see “How to decide which System Automation option to implement” on page 24). The configuration includes:

- four LPARs running z/OS in a sysplex
- a DB2 database with four data-sharing members
- two LPARs with z/VM containing a total of four Linux guests.

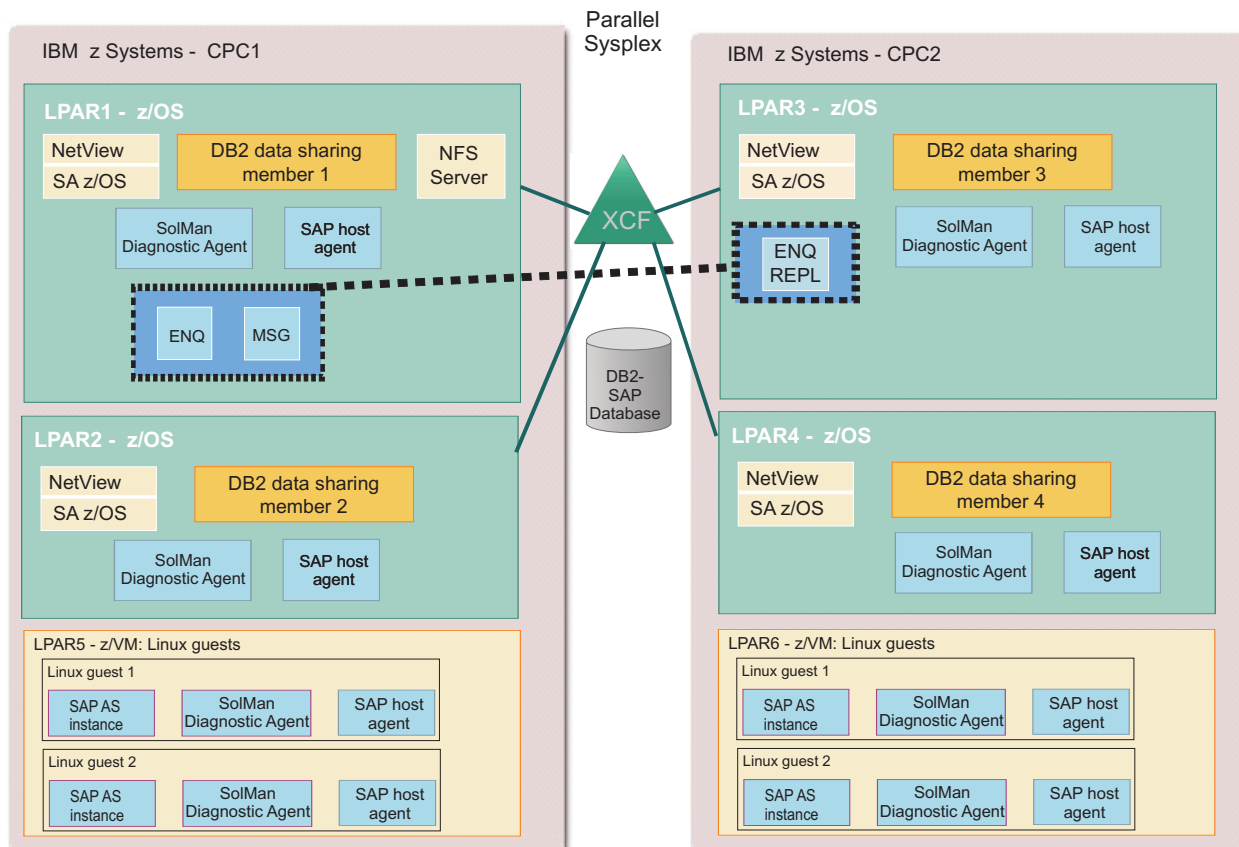


Figure 24. High availability solution configuration for SAP

Software prerequisites

This topic provides a summary of the software requirements.

Table 6 and Table 7 on page 91 describe the minimum level of software needed to implement the high availability solution for SAP, the recommended level of the software, and any special PTF requirements for each product. Be sure to check *SAP Note 81737* for the latest PTF requirements.

Table 6. Software requirements for the HA solution

Product name	Minimum level requirement	Recommended level
z/OS	Version 2.1	Version 2.2
DB2 Universal Database for z/OS	Version 11	Version 12
DB2 Connect	Version 11.1, special build 35734	Latest DB2 Connect Fixpack level that is certified by SAP. See <i>SAP Note 1927404: DB2-z/OS: IBM Data Server Driver for CLI/ODBC/JDBC/SQLJ - Special Builds</i>
Tivoli NetView for z/OS (required by System Automation for z/OS)	Version 6.1	Version 6.2 [1]
System Automation for z/OS	Version 3.4, APAR level OA44246	Version 3.5, APAR level OA51440

Table 6. Software requirements for the HA solution (continued)

Product name	Minimum level requirement	Recommended level
SAP Software Provisioning Manager	Latest SWPM Service Pack level available from the SAP Software Download Center . SWPM includes the latest level of <i>ZSCSinst</i>	
SAP WebDispatcher for z/OS UNIX	SAP Web Dispatcher Version 7.20 patch level 320	
All SAP NetWeaver solutions based on SAP 7.2x kernel [2].	SAP 7.21 kernel, patch level 623 and SAP 7.22 kernel, patch level 100	
SAP NetWeaver solutions based on SAP 7.4x kernel	SAP 7.42 kernel, patch level 425 and SAP 7.45 kernel, patch level 218	
[1]	Or higher supported level. NetView 6.x is required for SA z/OS 3.5.	
[2]	Older SAP 7.0x and 7.1x systems might have been installed with SAP's old profile structure (with separated start and instance profiles) and without a separate ERS instance. These systems must be converted to the new structure before they can automated with the <i>*SAPSRV add-on policy</i> (see Chapter 7, "Customizing SAP for high availability," on page 113).	

For SAP Netweaver 7.0, 7.1, 7.2, and 7.3, SAP has introduced a 7.2x downward compatible kernel (DCK) as a common kernel (see *SAP Note 2198998: SAP Kernel 7.22 disp+work (DW.SAR) patch forecast*).

For SAP Netweaver 7.4 and 7.5, SAP uses 7.4x kernels. For up-to-date information about SAP 7.4x kernel levels, see *SAP Note 1969546: Release Roadmap Kernel 740*.

Register for the following document in the SAP Community Network to obtain latest updates about SAP kernels: SAP Kernel: Important News. SAP kernels can be obtained from the **SAP Software Download Center**.

Table 7 lists operating system and automation software recommendations for SAP application servers.

Table 7. Software requirements for SAP application servers

Product name	Minimum level requirement	Recommended level
SAP application server on Linux on z Systems	SUSE Linux Enterprise Server 11 SP4 for IBM z Systems	SUSE Linux Enterprise Server 12 for IBM z Systems
SAP application server on AIX	AIX 6.1	AIX 7.1
SAP application server on Windows	Windows Server 2008 R2	Windows Server 2012 R2
Tivoli System Automation for Multiplatforms	4.1.0.3 [1]	4.1.0.3
z/VM	Version 5.4	Version 6.3
[1]	With SLES 12 it is mandatory to use the 64-bit version of Tivoli System Automation for Multiplatforms 4.1.0.1 or higher. Read "SA MP prerequisites when migrating to SLES 12" on page 315 about how to switch from earlier releases.	

Naming conventions

Certain conventions need to be selected for naming the policy components for Tivoli System Automation for z/OS and System Automation for Multiplatforms.

Recommendations for naming conventions exist in the following areas:

- "Naming conventions for Tivoli System Automation for z/OS" on page 92
- "Naming conventions used in the SA z/OS policy" on page 94

- “Naming conventions for Tivoli System Automation for Multiplatforms” on page 95

Naming conventions for Tivoli System Automation for z/OS

SAP recommends that you run one SAP system on one server. However, one of the strengths of z/OS is the ability to run components from *multiple* SAP systems on one z/OS Sysplex or even in one z/OS LPAR.

Each SAP system requires its own:

- ABAP and/or Java central services
- DB2 subsystem, with one or more DB2 data sharing members
- A set of file systems

Common questions that you might consider include:

- How do I monitor all SAP-related address spaces with SDSF?
- How can I efficiently use System Automation (SA) to manage SA resources and groups that belong to one specific SAP system.
- On which volumes should I allocate the 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 is clear that a good naming convention makes it easier to monitor and maintain each SAP system.

When you select the names for the components of one SAP system, IBM recommends that wherever possible you use the unique 3-character *SAP system identification* (denoted as <SAPSID> in Table 8 on page 93) as a part of the component names belonging to that SAP system. If you follow the naming convention you can use the SAP best practice policy and the SAP high availability wizard for SA z/OS with only minor post-processing.

It is recommended that you use **SAP<SAPSID>** or **S<SAPSID>** or just **<SAPSID>** as a prefix for all SAP resources which are related to a specific SAP system. Furthermore, it is recommended that you use **SAPSYS** as a prefix for all SAP resources which are *not related* to a specific SAP system

- Table 8 on page 93 lists the recommended names of all *z/OS-related components* of an SAP system, together with details of how or where they are defined.
- Table 9 on page 93 lists the recommended names of all *components of an individual SAP system* that are defined within SA z/OS.

Note: If you use BPX.DAEMON for program control (security) within your Sysplex, then each <sid>adm user must have read access to RACF Facility class BPX.JOBNAME. For detailed information, see *Tivoli System Automation for z/OS: Planning and Installation*.

The names that are used in the **Sample** column correspond to a sample SAP system **HA1** and a sample sysplex **COHPLEX**, which are used in many examples throughout this document.

Table 8. Recommended names for all z/OS-related components of an SAP system

Component	Recommended name	Sample	How/where defined
DB2 address spaces	<SAPSID>xMSTR	HA1xMSTR	PROCLIB member names
	<SAPSID>xDBM1	HA1xDBM1	
	<SAPSID>xIRLM	HA1xIRLM	
	<SAPSID>xDIST	HA1xDIST	
	<SAPSID>xLITE	HA1xLITE	
	(where x defines the data sharing (DS) member thru a unique character, for example 1 for DS one and 2 for DS two)		
High Level Qualifier for SAP VSAM objects	SAP<SAPSID>	SAPHA1	IDCAMS
High Level Qualifier for SAP zFS file systems:	(A) OMVS.ZFS.<SYSPLEX_NAME>.TRANS	(A) /usr/sap/trans ==> OMVS.ZFS.COHPLEX.TRANS	MOUNT FILESYSTEM command
(A) SAPSID independent	(B) OMVS.ZFS.<SYSPLEX_NAME>. <SAPSID>.USRSAP	(B) /usr/sap/HA1 ==> OMVS.ZFS.COHPLEX.HA1.USRSAP	
(B) SAPSID dependent	OMVS.ZFS.<SYSPLEX_NAME>. <SAPSID>.SAPMNT	/sapmnt/HA1 ==> OMVS.ZFS.COHPLEX.HA1.SAPMNT	
WLM definitions for service classes	<SAPSID>HIGH, <SAPSID>MED, <SAPSID>LOW	HA1HIGH, HA1MED, HA1LOW	WLM ISPF panels
NFS Server procedure name	MVSNFSHA	MVSNFSHA	PROCLIB member
VIPA name for ABAP SCS	<sapsid>ascsv	halascsv	TCP/IP DNS entry
VIPA name for Java SCS	<sapsid>scsv	halscsv	TCP/IP DNS entry
VIPA name for SAProuter (SAP system independent)	saproutev	saproutev	TCP/IP DNS entry
VIPA name for NFS server (SAP system independent)	sapnfsv	sapnfsv	TCP/IP DNS entry

Note:

1. A different naming convention might be necessary to allow the z/OS Storage administrator to use the data set name prefix as the SMS selector for the SMS Group or Storage or Data CLASS(es) or if the HLQ prefix is used in your installation to point to an SMS pool or to give a DATA CLASS that sets the correct zFS EXT attributes.
2. The second index level of <SAPSID> could allow a separate SMS POOL per SAP system (if required).
3. See Figure 25 on page 97 for the SAP directory structure and file system layout.

Table 9. Recommended names for all components of an individual SAP system

Component	Recommended name	Sample
Job name for ABAP SCS VIPA	SAP<SAPSID>ACV	SAPHA1ACV
Job name for ABAP enqueue server	<SAPSID><first 2 letters of instance name> <instance number><single digit number> [1]	HA2AS202

Table 9. Recommended names for all components of an individual SAP system (continued)

Component	Recommended name	Sample
Job name for ABAP enqueue replication server	<SAPSID><first 2 letters of instance name> <instance number><single digit number> [1]	HA2ER225
Job name for ABAP sapstart	<SAPSID><first 2 letters of instance name> <instance number><single digit number> [1]	HA2AS205
Job name for ABAP sapstartsrv	<SAPSID><first 2 letters of instance name> <instance number><single digit number> [1]	HA2AS209
Job name for ABAP message server	<SAPSID><first 2 letters of instance name> <instance number><single digit number> [1]	HA2AS201
Job name for Java SCS VIPA	SAP<SAPSID>JCV	SAPHA1JCV
Job name for Java enqueue server	<SAPSID><first 2 letters of instance name> <instance number><single digit number> [1]	HA2SC215
Job name for Java enqueue replication server	<SAPSID><first 2 letters of instance name> <instance number><single digit number> [1]	HA2ER231
Job name for Java message server	<SAPSID><first 2 letters of instance name> <instance number><single digit number> [1]	HA2SC214
Job name for Java gateway	<SAPSID><first 2 letters of instance name> <instance number><single digit number> [1]	HA2SC216
Job name for Java sapstart	<SAPSID><first 2 letters of instance name> <instance number><single digit number> [1]	HA2SC213
Job name for Java sapstartsrv	<SAPSID><first 2 letters of instance name> <instance number><single digit number> [1]	HA2SC217
Job name for SAP host agent (SAP system independent)	SAPHEXE<single digit number> [1]	SAPHEXE5
Job name for SAProuter (SAP system independent)	SAPSYSRT	SAPSYSRT
<p>[1] <single digit number> is a number chosen by UNIX System Services when creating job names. See http://www.ibm.com/support/knowledgecenter/SSLTBW_2.2.0/com.ibm.zos.v2r2.bpxb200/jomvs.htm.</p> <p>In order to map an actually running job to the exact SAP instance process, run the D OMVS,U=<SAPSID>ADM command for the LPAR where the instance processes are active.</p>		

Naming conventions used in the SA z/OS policy

A table contained in this topic summarizes the naming conventions used for the SA z/OS policy.

This policy is described in “How to adapt the SA z/OS *SAPSRV add-on policy” on page 149.

Table 10. Naming conventions for SA z/OS resources

Type of resource	Naming convention
Groups with sysplex scope	*X
Names for SAP resources related to SAP system <SAPSID>	SAP<SAPSID>*

Table 10. Naming conventions for SA z/OS resources (continued)

Type of resource	Naming convention
Names for SAP infrastructure resources	SAPSYS* and for the sapstartsrv groups of each SAP instance S<SAPSID>*

Naming conventions for Tivoli System Automation for Multiplatforms

This information summarizes the naming conventions for the SAP be used in the Tivoli System Automation for Multiplatforms policy.

For new installations, it is recommended to generate an SA MP policy that is based on the *SAP high availability policy feature* of System Automation for Multiplatforms 4.1.0.3 or higher (see Chapter 9, “Customizing System Automation for Multiplatforms,” on page 185). The generated SA MP resources follow the naming convention that is listed in the *Tivoli System Automation for Multiplatforms Version 4 Release 1 High Availability Policies Guide* (SC34-2660-02), which is available from the *IBM Knowledge Center for Tivoli System Automation for Multiplatforms*. If you continue to use an SA MP policy for SAP that was created from the sample in `zSAP_BusinessContinuity.zip`, refer to earlier editions of this guide for a list of naming conventions for the SAP resources.

Setting up DB2

In a high availability environment, there is little sense in making SAP Central Services (SCS) highly available if the database server is not highly available. For this reason you should use DB2 data sharing.

Because SCS does not connect to the database, there is no technical requirement to install it in one of the LPARs containing a DB2 subsystem, although it is possible to do so.

Using the **DB2 add-on policy* to perform a light restart

In case of an LPAR or DB2 member failure, a restart of the DB2 member is needed to allow for recovery. The **DB2 add-on policy* that is shipped with SA z/OS, automates this restart in the following way:

- *Normal* restart in place, if the LPAR is available.
- *Light* restart on another LPAR in case of a failure of the original LPAR.

The LIGHT option of the START DB2 command performs the following:

1. Restarts a DB2 data sharing member with a minimal storage footprint.
2. Terminates the DB2 member normally after freeing any retained locks that were held by this member.

Setting up file systems under z/OS for local and NFS exported file systems

Shared directory access between z/OS systems is required to allow the failover of the SAP Central Services instances running on z/OS. It is also needed for the movable NFS server.

For performance and availability reasons it is highly recommended that you use shared zFS file systems and run zFS in sysplex-aware mode. zFS file systems allow you to define shared as well as system-specific file systems by using special variables in the path name. If all your SAP systems are within one Sysplex, you can share all the files. For example, if you have one production Sysplex and one test Sysplex, and still want to use the same file systems, for example, the common SAP transport directory `/usr/sap/trans`, you must use the NFS Server/Client feature.

File systems

Read the recommendations about the location of the involved file systems to achieve a high availability.

For availability reasons it is recommended that you place the following SAP global directories on z/OS and share (export) them via the NFS server on z/OS with any non-z/OS hosts on which the SAP application servers run:

- `/sapmnt/<SAPSID>/exe`
- `/sapmnt/<SAPSID>/profile`
- `/sapmnt/<SAPSID>/global`
- `/sapmnt/<SAPSID>/trans`

In SAP 7.10 and subsequent SAP releases, the SAP installation copies the SAP kernel executable routines into a different directory to the one used in previous releases (SAP 7.00 and earlier). The directory name now reflects the platform for which the kernel routines were built. For example, the runtime directory of the SAP 7.10 application server for Linux on z Systems is:

```
/sapmnt/<"SAPSID">/exe/uc/linuxs390x
```

The mount point on the remote application server kernel routines for SAP 7.10 and higher remains:

```
/sapmnt/HA1/exe
```

File systems and directory structure

Figure 25 on page 97 shows the SAP directory structure and file systems of an SAP NetWeaver PI high availability installation with the ABAP and Java stack. SAP high availability means that every instance can run on a separate host. Figure 25 on page 97 also shows which zFS file systems must be defined to run ASCS/SCS and their ERS under z/OS UNIX and to automate them by SA z/OS. It also depicts which directories are exported by the NFS server to the application servers on AIX or Linux.

Note: Windows requires a different file system setup and directory structure, because there must be just one file system under z/OS UNIX System Services (z/OS UNIX) on the mainframe that contains all data of the SAP system to be installed. See the SAP SCN article: SAP HA Installations on z/OS and Windows Application Servers.

SAP directory structure and file systems (SAP 7.10 and higher)

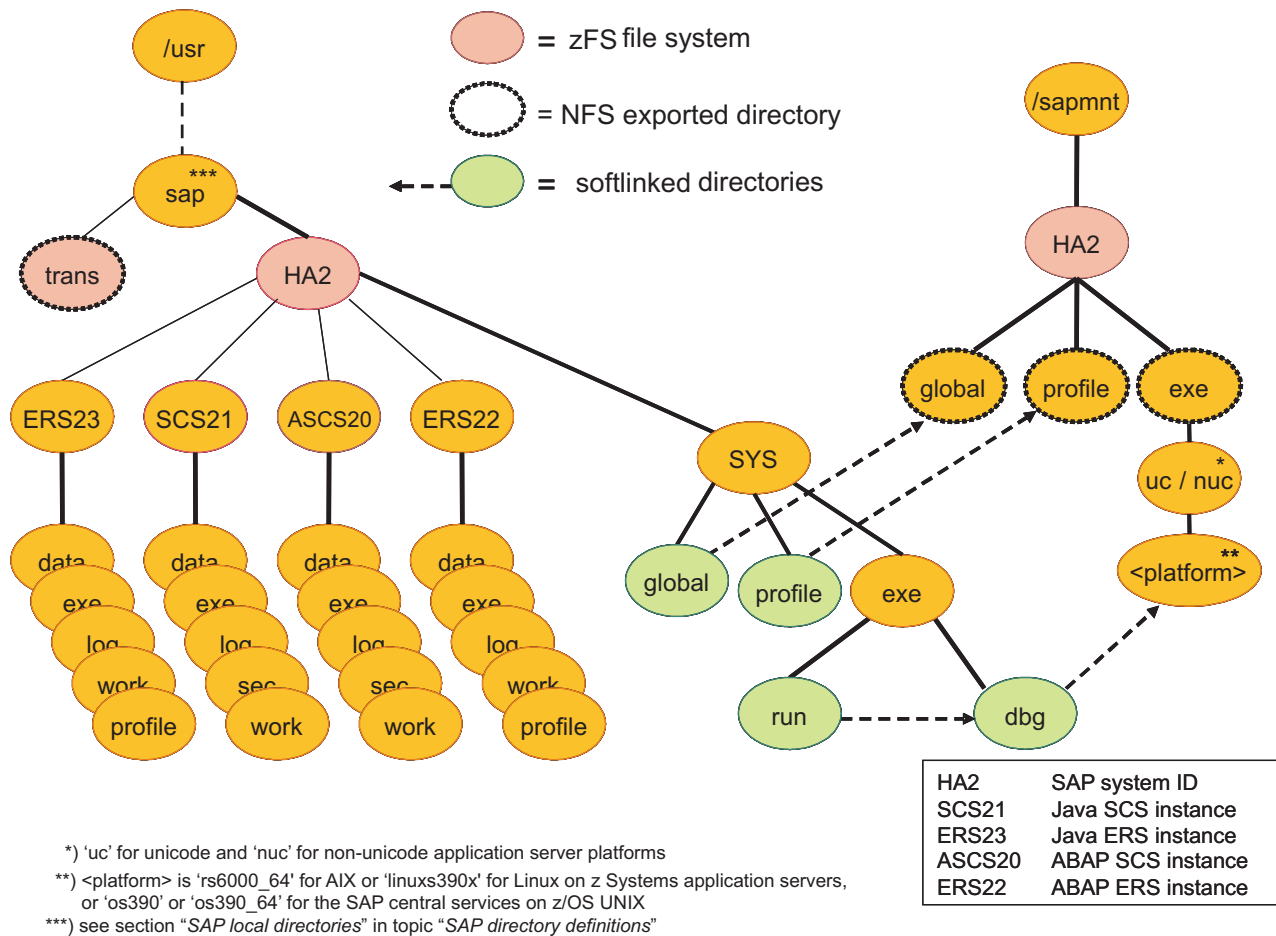


Figure 25. SAP directory structure and file systems

Setting up zFS file systems

This information provides recommendations how to set up zFS file systems to achieve high availability and to improve file system access and performance.

For high availability, it is recommended that:

- zFS file systems are allocated from a DFS SMS DASD POOL
- the SMS DATACLASS associated with all zFS instances is defined as follows:
 - DSN TYPE=EXT, PREFERRED, EXTENDED-ADDRESSABILITY
 - The dynamic volume count should be high enough to spread the workload and provide for future space requirements
- The zFS option aggrgrow is set to ON as a default. Use the following command to verify this:

```
zfsadm configquery -aggrgrow
```

These recommendations allow the zFS systems to expand on demand if sufficient volumes and space are available within the associated DFS SMS DASD POOL.

After the SMS customization is complete, you can go ahead with creating zFS file systems. Here are some examples of how to create a zFS for the SAPMNT subdirectory /sapmnt:

- `zfsadm define -aggregate OMVS.ZFS.COHPLEX.SAPMNT -megabytes 5000 1000`. This command creates a z/OS VSAM LDS named OMVS.ZFS.COHPLEX.SAPMNT of primary size 5 GB, and secondary size of 1 GB. **Note:** If not resolved via DSN, SMS options such as data, management, and storage class can also be specified.
- `zfsadm format -aggregate OMVS.ZFS.COHPLEX.SAPMNT -owner ha1adm -group sapsys -compat`. This command formats the new file system in compatibility mode and associates an owner and group to it.
- `/usr/sbin/mount -t zfs -a yes -f 'OMVS.ZFS.COHPLEX.SAPMNT' /sapmnt`. This command (temporarily) mounts the new file system on the mount point /sapmnt.

Note:

1. The AUTOMOVE=YES MOUNT option is required for high availability in the event of an LPAR outage.
2. Permanent mounting of any z/OS UNIX file systems should be done through BPXPRMnn members of SYS1.PARMLIB.
3. Correct SPACE requirements can be determined from the SAP installation manuals, and remember that you might require additional space for backups especially when adding new maintenance.
4. Further information about the usage of the zFS file system type can be found in *z/OS Distributed File Service zFS Administration*.
5. It is recommended to enable zFS *sysplex-aware* support for performance improvements and for availability of the NFS exported zFS file systems (see subsequent sections in this topic).

Using sysplex-aware file systems

For optimal file system access performance, it is recommended that you create all SAP file systems as *sysplex-aware*.

In a shared file system environment, z/OS V1R13 or later versions are always enabled to allow zFS read/write file systems being sysplex-aware (zFS runs `sysplex=filesys`). You can individually choose which file systems are sysplex-aware for read/write and which ones are not. The default is that zFS read/write file systems are not sysplex-aware. A newly mounted zFS read/write file system is made sysplex-aware by specifying the RWSHARE mount parameter, as shown:

```
MOUNT FILESYSTEM('OMVS.ZFS.COHPLEX.SAPMNT') TYPE(ZFS) MOUNTPoint('/sapmnt')  
PARM('RWSHARE')
```

As an alternative, you can specify `sysplex_filesys_sharemode=rwshare` in your IOEFSPRM. This changes the default and has the effect that zFS read/write file systems are mounted sysplex-aware (unless you explicitly specify the NORWSHARE mount parameter).

z/FS file system performance improvements with large directories

zFS file systems that currently exist on z/OS V1R13 and earlier systems, are version 1.4 aggregates. These version 1.4 aggregates contain v4 directories. If you are experiencing poor performance with large v4 directories, consider using the version 1.5 aggregates with extended (v5) directory support that is available

starting with z/OS V2R1. See z/OS Distributed File Service zFS Administration, SC23-6887 for information regarding v4 directory considerations and guidelines, for converting aggregates and directories from v4 to v5, as well as for using version 1.5 aggregates and extend (v5) directories.

Verify your zFS settings

To verify your current zFS settings, issue the command: `F DFSZFS,QUERY,LEVEL`. This returns an output similar to the following:

```
IOEZ00639I zFS kernel: z/OS zFS
Version 02.02.00 Service Level 0A50873 - HZFS420.
Created on Fri Sep 16 14:28:08 EDT 2016.
sysplex(filesys,rwshare) interface(4)
```

SAP directory definitions

Read the topic for information about the SAP directories that need to be defined.

SAP global transport directory

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

SAP system-wide directories

The subdirectories of `/usr/sap/<SAPSID>/SYS` are defined at installation time as symbolic links to the corresponding subdirectories of `/sapmnt/<SAPSID>`, for example, `/usr/sap/HA1/SYS/profile` points to `/sapmnt/HA1/profile`. The directory `/sapmnt` must be created in the root file system and shared in the sysplex.

The directory `/sapmnt/<SAPSID>` is the mount point for the SAP system-wide file system. This file system must 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 `$VERSION/usr`. This means that the contents of the `/usr` directory are different on every LPAR. However, it is not practical for the `/usr/sap` directory. It is recommended that you create the directory `/sap` in the root file system and to define symbolic links for `/usr/sap` to point to `/sap`. The symbolic links must be defined on each LPAR, that is, in each `$VERSION/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 point `/usr/sap/<SAPSID>` for the instance-specific file systems, such as `ASCS00`. These file systems do not need to be exported by NFS.

In order to get best performance in a shared zFS environment, all SAP file systems should be configured *sysplex-aware* (see "Using sysplex-aware file systems " on page 98).

SAP administrator home directory

The home directory `/u/<sid>adm` is shared in the sysplex.

SAP host agent file system

SAP requires to run an SAP host agent in each z/OS LPAR on which SAP components can run. One of the reasons for this is that the SAP host agent is used to collect and deliver data which is only locally available on each LPAR. The SAP host agent infrastructure on each LPAR is required by other SAP tools, like for example, the *SAP NetWeaver Landscape Virtualization Management (LVM)*.

During installation of an SAP Central Services instance, the installation procedure places the SAP host agent into the SAP system independent path `/usr/sap/hostctrl`. SAP additionally requires that this path is local to the machine where the SAP host agent runs. On the other side, z/OS UNIX zFS file systems are normally shared between all the LPARs in a SYSPLEX. This is especially true for the `/usr/sap` directory, which is a link to `/sap`.

In order to have zFS file systems locally, you must take specific actions. In the case of an SAP host agent, the following recommendations are given. Note that the SA z/OS resources, which are part of the **SAPSRV add-on policy*, exactly assume such a recommended setup.

The first step is to create a local zFS file system on each LPAR, for example like *saphostctrl*. The following samples show only the steps for one LPAR called COH1:

```
COH1:vsch:/COH1/saphostctrl>df -kvP .
Filesystem 1024-blocks Used Available Capacity Mounted on
OMVS.COH1.HOSTCTRL 360000 158294 201706 44% /COH1/saphostctrl
ZFS, Read/Write, Device:84, ACLS=Y
File System Owner : COH1 Automove=N Client=N
Filetag : T=off codeset=0
Aggregate Name : OMVS.COH1.HOSTCTRL
```

The second step is to copy the contents of the shared zFS, which was used during installation of the SAP host agent, into the new local zFS file system on each LPAR:

```
COH1:vsch:/COH1/saphostctrl>cp -p -R /usr/sap/hostctrl/*
```

The third step is to delete the standard shared zFS and to create a link to the local copies:

```
cd /usr/sap/hostctrl
rm -R *
cd ..
rmdir hostctrl
ln -s /\$SYSNAME/saphostctrl hostctrl
```

Now you have a *hostctrl* -> */\$SYSNAME/saphostctrl* link in the shared `/usr/sap` directory (which is a link to `/sap` directory). This way the *saphostctrl* is a local directory on COH1. This procedure needs to be repeated for each LPAR in your sysplex.

The consequence of having the `hostctrl` directory locally to each LPAR is that you must care for manually propagating adaptations/changes/updates to each LPAR, where the SAP host agent can run.

The following automated mechanism can be used to automatically propagate updates of the SAP host agent executables to all LPARs from one central shared directory.

For details on the automated upgrade see topic *Automated Upgrade of SAP Host Agent* in the SAP Host Agent installation documentation (available as PDF attachment to *SAP Note 1907566: Obtaining the Latest SAP Host Agent Documentation*).

For example, you might create a directory `hostctrl_update` in the shared file system `/usr/sap` and configure the automated upgrade of the SAP host agent to use this directory by setting **DIR_NEW** in the host agent profile **host_profile** to point to this directory:

```
DIR_NEW = /usr/sap/hostctrl_update
```

The SAP host agent restarts itself as the last step of the automated upgrade. System Automation recognizes this restart but will not trigger a restart of its own, provided the automatic restart of **saphostexec** is fast enough.

If you want to completely avoid any interaction between the automated upgrade and System Automation, you should disable automation for the SAP host agent before copying the new SAP host agent code level into **DIR_NEW**. Do not forget to re-enable automation after the SAP host agent successfully restarted itself.

See the SAP documentation for further configuration options that help you to avoid incomplete updates and that influence the time when the automated upgrade is triggered.

NFS server on z/OS

Read this information to learn about various aspects on the installation of an NFS server on z/OS. It is recommended that you run one dedicated NFS server as this greatly simplifies the setup and future diagnostics.

It is highly recommended that you read topic *NFS Server on z/OS* in the *Planning Guide for SAP on IBM DB2 for z/OS* to learn what is necessary to setup a standard NFS Server under z/OS. This *Planning Guide* is available from the *SAP on DB2 for z/OS* community at <https://go.sap.com/community/topic/db2-for-zos.html>. Then, from the **Key Topics** section, select *SAP Information Sources*.

A highly available z/OS NFS Server setup has additional requirements. For example, the mount handle database must be accessible from all LPARs, where the NFS Server can run on. Also, the start procedure with the `moddvipa` statement to activate the NFS Server VIPA, must be accessible on all those LPARs.

It must be possible to move the z/OS NFS server used by SAP high availability NFS clients between z/OS LPARs within the same SYSPLEX. The associated dynamic VIPA can only be moved within the same TCPIP subplex.

It is recommended to start the associated dynamic VIPA within the start procedure of the NFS server using the `MODDVIPA` utility. Add the following as a first step to your NFS start procedure, using your NFS dynamic VIPA IP:

```
//TCPDVP EXEC PGM=MODDVIPA,REGION=0K,TIME=1440,  
//PARM='POSIX(ON) ALL31(ON)/-p TCPIP -c <dynamic_VIPA>'
```

PARMLIB member `BPXPRMxx` must contain the keyword `SYSPLEX(YES)` for the SAP z/OS UNIX System Services file systems to be accessible from multiple z/OS LPARs within a SYSPLEX.

For the NFS client running on the SAP application server the movement of the NFS server is transparently handled via a z/OS dynamic VIPA, dynamic routing (usually OSPF), and their automatic reconnect ability.

NFS server security model

It is highly recommended that you use `security(exports)`. This allows NFS clients to transparently connect or reconnect to NFS file systems even when there are NFS server failures and restarts across different z/OS hosts.

Transparent connect is possible, because `security(exports)` does not require any RACF definitions or the use of the `mvslogin` command. It is the default UNIX NFS server mode of operation.

The export list for the movable NFS server is limited to global SAP directories, which do not contain sensitive data. Access is also limited to specific client IP addresses only.

For further information about setting up NFS, see *Network File System Customization and Operation*.

Mount handle databases and the remount site attribute

To allow transparent failover of the NFS server, the mount handle databases must be shared between the z/OS hosts.

If your NFS clients use protocol version 3 to connect to the z/OS NFS Server, you must use the `remount` attribute. In an NFS client/server setup where protocol version 4 only is used, it is not necessary to set the `remount` attribute because most NFS version 4 clients handle volatile file handles automatically.

For information about how to upgrade from NFS Version 3 to NFS Version 4, see “NFSv4 migration hints and tips” on page 302.

The nlm site attribute

If you have NFS version 3, you must make sure that the NFS Lock Manager (NLM) is started on the z/OS NFS server. Add the `nlm` attribute to the NFS server attributes file.

NLM creates TCP connections from z/OS to the NFS clients. Because you must use a dynamic VIPA for the high availability NFS server, this dynamic VIPA must be a source VIPA. This can be configured in the z/OS TCP PROFILE(s) by using the `SRCIP` statement. See “z/OS VIPA usage for the high availability solution for SAP” on page 58. Read the section about NLM and z/OS NFS server and set up your environment accordingly.

Note: If you have NFS version 4 clients only, it is not necessary to set the `nlm` site attribute or to use source VIPA because NFS version 4 protocol handles locking automatically.

The restimeout site attribute

`restimeout(n,m)` specifies a retention period and a clock time for the removal of mount points and control blocks that have been inactive longer than the specified retention period.

If `n` is set to 0, the z/OS NFS server does not remove any mount points or associated resources.

It is recommended that you specify `restimeout(720,0)` (30 days) or `restimeout(0,0)`, which means no timeout.

Moving the ownership of z/OS NFS server exported z/OS UNIX file systems to a specific LPAR

For best performance of the NFS server exported file systems, these file systems must be defined as sysplex-aware zFS. If the NFS server is (re)started for whatever reason on another LPAR by SA z/OS, zFS then automatically moves the zFS ownership of a zFS file system to the system that has the most read/write activity. When all exported zFS file systems are defined as sysplex-aware, then no further action is required to move the file system ownership when the NFS server moves to another LPAR.

The configuration of a sysplex-aware zFS is described in *Distributed File Service zFS Administration*.

NFS client root access

The SAP installation on a remote (AIX, Linux, or Windows) application server requires root (`uid=0`) access to all NFS-mounted SAP file systems. The z/OS NFS server exports file must specify the NFS clients (by name or IP) and also use the suffix option `<root>`

For example, to allow NFS clients using VIPAs of 10.101.4.214, 10.101.4.215, and 10.101.4.216 to have read and write root access to the SAP profile subdirectory, you enter:

```
/hfs/sapmnt/HA1/profile -access=10.101.4.214<root>|\
                        10.101.4.215<root>|\
                        10.101.4.216<root>
```

Note: z/OS UNIX System Services file system path names for a mount path directory must be prefixed. The prefix `/hfs` in the exports file is the z/OS system default prefix. It does not indicate that the exported directory must reside in an HFS filesystem. In fact, it is recommended to use zFS file systems for all SAP related file systems in z/OS UNIX. If you want to change the default prefix, you can do so by setting the NFS server attribute `HFSPREFIX` (for details see *z/OS Network File System Guide and Reference*).

NFS clients general information

This topic provides mounting information for NFS directories and naming convention information to ensure a functioning set up of an NFS client - server environment.

Before starting the set up an NFS client, you should read topic *NFS Clients* in the latest *Planning Guide for SAP on IBM DB2 for z/OS* from <https://go.sap.com/community/topic/db2-for-zos.html> to learn what is necessary to set up an NFS client in general, and specifically on supported SAP application server platforms.

It is highly recommended that you set up the automount daemon to mount SAP NFS directories. This way you get automatic mounts and remounts of the SAP NFS directories via the operating systems infrastructure.

For NFS clients that connect to the z/OS NFS server it is recommended that you store all data in EBCDIC code page 1047. For an example about how to set up the automount daemon and code-page translation, see “NFS client automount samples” on page 301.

All mounts must use the name of the Dynamic VIPA associated with the z/OS NFS server. This allows mounts and remounts to function correctly if the z/OS NFS server is moved from one LPAR to another.

NFS clients, which run in addition to the recommended highly available network setup, must ensure that they have a local source VIPA set, so that all IP packets sent to the z/OS NFS server always appear to come from the same host regardless of the local interface used. This is important because the z/OS NFS server always associates mounts, remounts, and general NFS file I/O handles to the source IP address. If the NFS client's source IP should change, then the z/OS NFS server may consider each IP as a separate client (depending on DNS records for each client).

NFS clients must also ensure that any user name and group name they use to access NFS files are associated with exactly the same **uid** and **gid** as on the z/OS NFS Server.

Non z/OS NFS server

If you plan to run the NFS server for your SAP system under Linux on z Systems or AIX, you must make that NFS server highly available.

Tivoli System Automation for Multiplatforms offers a specialized NFS server HA policy for this purpose. See topic *NFS high availability policy* in the *Tivoli System Automation for Multiplatforms Version 4 Release 1 High Availability Policies Guide* (SC34-2660-02), which is available from the *IBM Knowledge Center for Tivoli System Automation for Multiplatforms*.

Configuring Sysplex performance monitoring highly available with the RMF Distributed Data Server (RMF DDS)

Read this topic to learn about how to use the RMF DDS in order to obtain a highly available performance monitoring of all systems in the Sysplex where your SAP environment is running.

Applications that want to access Sysplex-wide performance data can use the RMF Distributed Data Server (RMF DDS) to retrieve that data from a single data server on one system in the Sysplex. For information on how to configure and start the DDS, and about its benefits, read the *z/OS RMF User's Guide* and the *z/OS RMF Programmer's Guide*.

The SAP components `sapcimb` (see “SAP host agent” on page 142) and the `sysplex monitor` function, which is part of the SAP transaction **DBAcockpit**, connect to the RMF DDS to retrieve z/OS performance data.

You can choose one of the following methods to make use of the performance monitoring that is provided by the RMF DDS:

- With **Tivoli System Automation**, you can use the resource definitions for RMF and RMFGAT, which are contained in the **BASE add-on policy* of SA z/OS. Starting and stopping of the RMF DDS is then managed automatically by SA z/OS.

- Without **Tivoli System Automation**, follow the information in topics *Setting up the Distributed Data Server for z/OS* and *Starting the Distributed Data Server* in the *z/OS RMF User's Guide*.

In addition to the automated starting and stopping of the RMF DDS, you need a VIPA that instantly follows the RMF DDS in case it moves to another system in the Sysplex. In the procedure described hereafter, we use a non-disruptive, distributed dynamic VIPA (DVIPA).

Setting up the DVIPA

In your TCP/IP configuration, you need to add a VIPA range definition like this:

```
VIPARANGE DEFINE MOVEABLE NONDISRUPTIVE 255.255.255.248 9.152.20.176
```

Publish the DVIPA in your DNS with an entry like 9.152.20.177 <DDS hostname> where <DDS hostname> is your choice. Together with the value DOMAINORIGIN from /etc/resolv.conf, this entry combines to the full qualified name for the RMF DDS host. This name is required in a later step of this procedure (see "Configure the sysplex monitor function in SAP DBACOCKPIT"). With this prerequisite, the RMF DDS can be reached by its IP address, by its short-name <DDS hostname>, and by its long name <DDS hostname>.DOMAINORIGIN.

Bind together the DVIPA with the RMF DDS

In your TCP/IP configuration, you need to add a port bind statement. If for example, the RMF DDS listens on port 8803, then you need to specify an entry similar to the following one:

```
8803 TCP RMFDDS01 BIND 9.152.20.177; VIPA FOR RMF DDS FOR SAP SYSPLEX MON
```

Configure the CIM provider component of the SAP host agent

To enable performance monitoring by the CIM provider component (sapcimb) of the SAP host agent, you need to configure access to the RMF DDS. Set the environment variables SAP_IBMZMON_DDS_HOST to the DVIPA host name and the SAP_IBMZMON_DDS_PORT to the DDS port. For example, add statements similar to the following to the .cshrc file of your saproot user:

```
setenv SAP_IBMZMON_DDS_HOST osmonitorv
setenv SAP_IBMZMON_DDS_PORT 8803
```

For more details on sapcimb configuration for z/OS, see *SAP Note 2047924: DB2-z/OS:CCMS:HAG: CIM-Provider Enablement for z/OS*.

Configure the sysplex monitor function in SAP DBACOCKPIT

1. Logon to your SAP GUI and call transaction **DBACOCKPIT**.
2. Navigate to: **Configuration -> OS Monitoring Settings**
3. Edit the parameters for the RMF Distributed Data Server (DDS): Hostname = <DDS name>, or IP address, or <DDS name>.DOMAINORIGIN. Port = 8803
4. Check **Enable OS07n data collection**.
5. Click the Save symbol.

A test connection is executed. A successful test is indicated by the message: Test connect successful.

Verification

First use SA z/OS to stop and start the APLs RMF and RMFGAT. As an extra test, stop or cancel the address spaces of RMF, RMFGAT, and RMFDDS01 manually.

During your tests, you can use either the DBA Cockpit of SAP, the Sysplex Monitor, or the RMF Data Portal on <http://<DDS name>:8803> to verify that performance data is continuously available.

Result

Any time that you stop or cancel RMF, or RMFGAT, or RMFDDS01, the RMF DDS is stopped and restarted on the current RMF master LPAR. The VIPA automatically moves along with the RMF DDS. The Sysplex Monitor data is continuously available.

Modifying the environment for AIX and for Linux on z Systems

It is recommended that you run the highly available NFS server on z/OS, using OSPF as a dynamic routing protocol for network high availability (network between application server and database server) and run automounter to mount the NFS file systems. Make the following changes to your environment so that all of these work together smoothly at AIX and Linux startup or shutdown time.

The required scripts mentioned in the contained subtopics, `rc.local` for AIX and `testNFS` for Linux, are included in the `zSAP_BusinessContinuity.zip` file. For details, see “Automation scripts” on page 285.

Because the processes that implement the OSPF protocol (`ospfd/zebra/quagga` under Linux and `gated` under AIX) need some time to create a routing table after they are started and the normal boot sequence does not allow “waiting” for successful routing table updates, you must adapt some of the services called during startup and shutdown.

Important: If you change any script that is delivered with the operating system (such as `/etc/rc.nfs`), the changes can get lost or overwritten during a service update. This means that you must check the adapted scripts after such activities.

Modifying the environment for AIX

The topic contains the documentation of the procedure that is required to adapt your environment for AIX.

About this task

Make the following changes to your AIX 6.1 system environment. In `/etc/rc.nfs` remove the call to start the automounter as you want to run the *Business Continuity for SAP on IBM z Systems* automounter startup script, which waits up to three minutes until the AIX component `gated` has added a route to the NFS Server. Otherwise, the automount is automatically aborted and never recovered.

Procedure

1. Comment out the start of automount in `/etc/rc.nfs`, as in this example:

```
## if [ -s /etc/auto_master ]; then
##     /usr/sbin/automount
## fi
```

2. Add a new rc.local script in the /etc directory. Adapt the virtual NFS server host name to your environment (the virtual NFS server host name in this procedure is sapnfsv):

```
$ cat /etc/rc.local
#!/bin/ksh
# this is to start autofs group after gated had time to consolidate it's routing table
VirtNFSHOST="sapnfsv"
TIMEVAR=0
MAXRETRY=3 # this lets this script try for 3 minutes
RETRY=0
SLEEPSEC=1 # this lets this script sleep for 1 seconds
RC=1

while [[ $RC -eq 1 && $RETRY -lt $MAXRETRY ]];do
    while [[ $RC -eq 1 && $TIMEVAR -lt 60 ]];do
        eval "/usr/bin/netstat -r | /usr/bin/grep \"\$VirtNFSHOST\" > /dev/null"
        RC=$?
        TIMEVAR=$(( ${TIMEVAR} + ${SLEEPSEC} ))
        sleep ${SLEEPSEC}
    /usr/bin/logger -p user.debug -i "$0: Time to wait for NFS server $VirtNFSHOST in routing table is $TIMEVAR sec."
    done
    if [[ $RC -eq 0 ]];then
        /usr/sbin/automount
    /usr/bin/logger -p user.debug -i "$0: Found (virt.) NFS server $VirtNFSHOST in routing table; automount daemon started."
    else
        if [[ $RETRY -eq $((${MAXRETRY}-1)) ]];then
            /usr/bin/logger -p user.debug -i "$0: Could not find (virt.) NFS server
            $VirtNFSHOST in routing table; automount daemon not started."
        else
            /usr/bin/logger -p user.debug -i "$0: Could not find (virt.) NFS server $VirtNFSHOST in routing table; Retrying ..."
        fi
        TIMEVAR=0
        RETRY=$(( ${RETRY} + 1 ))
    fi
done
exit 0
```

3. Add starting of rc.local to /etc/inittab. Insert the rc.local call before the rc 2 call:

```
automd:2:wait:/etc/rc.local >/dev/console 2>&1
l2:2:wait:/etc/rc.d/rc 2
```

Note: If you start the sapstartsrv processes in /usr/sap/sapservices, some NFS mount failure messages are generated in the /tmp/syslog.out file at system start. This is normal because even if the z/OS NFS server host is listed in the routing table, it might not be able to reply directly because the z/OS routing daemon also needs to update its routing table after the AIX (gated) is started.

Modifying the environment for Linux on z Systems

This topic contains the documentation of the procedure that is required to adapt your environment for Linux on z Systems. You must set dependencies for the initialization scripts sapinit and autofs so that they are started and stopped in the correct order.

sapinit needs a start dependency to autofs and autofs needs a start dependency to ospfd/zebra. Therefore, in the sapinit script under /etc/init.d at the end of the # Required-Start line, add the autofs:

```
# Copyright (c) 1995-2005 SAP AG Walldorf, Germany.
#
# /etc/init.d/sapinit
#
# chkconfig: 345 90 10
# description: Start sapstartsrv
#
```

```
### BEGIN INIT INFO
# Provides: sapinit
# Required-Start: $network $syslog $remote_fs $time autofs
```

You must set dependencies for the init scripts sapinit and autofs so that they are started and stopped in the correct order.

In the autofs script under /etc/init.d, add the **ospfd** and **zebra** processes at the end of the # Required-Start line:

```
ihlscohl:/etc/init.d # head -n 20 autofs
#!/bin/bash
#
# rc file for automount using a Sun-style "master map".
# first look for a local /etc/auto.master, then a YP
# map with that name
#
```

```
### BEGIN INIT INFO
# Provides:      autofs
# Required-Start: $network $syslog $remote_fs zebra ospfd
```

Note that with SLES 12, init.d has been replaced by systemd. Read “Tips for migrating from SLES 11 to SLES 12” on page 313 to see how this dependency and the call of the testNFS script is done under SLES 12.

Additionally, add the following four lines marked **bold** into the start section of the script (/etc/init.d/autofs):

```
case "$1" in
start)
# changes for SAP
echo
echo "Info: Starting program testNFS to check if NFS server IP is reached via ping"
/root/testNFS
echo -n "Starting $prog "
```

The testNFS script waits up to three minutes until the NFS server VIPA can be pinged. Copy the testNFS script with correct permissions into the correct directory, in this example here, it is into /root. Adapt the virtual NFS server host name to your environment (in this example, the (virtual) NFS server host name is sapnfsv):

```
cat /root/testNFS
#!/bin/ksh
# this is to start autofs group after ospf/zebra had time to consolidate it's routing table
VirtNFSSHOST="sapnfsv"
TIMEVAR=0
MAXRETRY=3 # this lets this script try for 3 minutes
RETRY=0
SLEEPSEC=1 # this lets this script sleep for 1 seconds
RC=1

while [[ $RC -eq 1 && $RETRY -lt $MAXRETRY ]];do
    while [[ $RC -eq 1 && $TIMEVAR -lt 60 ]];do
        eval "/bin/netstat -r | /usr/bin/grep \"\$VirtNFSSHOST\" > /dev/null"
        RC=$?
        TIMEVAR=$(( ${TIMEVAR} + ${SLEEPSEC} ))
        sleep ${SLEEPSEC}
        /bin/logger -p user.debug -i "$0: Time to wait for NFS server $VirtNFSSHOST in routing table is $TIMEVAR sec."
        done
        if [[ $RC -eq 0 ]];then
            /bin/logger -p user.debug -i "$0: Found (virt.) NFS server $VirtNFSSHOST in routing table; automount daemon can be started."
        else
            if [[ $RETRY -eq $(( $MAXRETRY - 1 )) ]];then
                /bin/logger -p user.debug -i "$0: Could not find (virt.) NFS server $VirtNFSSHOST in routing table; "
                "$0: automount will fail to mount NFS mounts..."
            else
                /bin/logger -p user.debug -i "$0: Could not find (virt.) NFS server $VirtNFSSHOST in routing table; Retrying ..."
            fi
        fi
    done
done
```

```
TIMEVAR=0
RETRY=$((RETRY)+1)
fi
done
exit 0
```

Recommendations for sysplex failure management (SFM) policy definitions

Read this topic to learn how to set up system status detection (SSD) and the sysplex failure management (SFM) to enable fast detection of LPAR outages.

To partition an LPAR that failed, or was manually reset, or was re-IPLed fast and automatically without operator intervention, it is recommended that you exploit system status detection (SSD) and set up a sysplex failure management (SFM) policy with the following default values according to *z/OS MVS Setting Up a Sysplex*.

- Use the default value of YES for the CONNFAIL parameter. If you specify NO, this indicates that the system is to prompt the operator to decide which system or systems to partition from the Sysplex whereas the default value does not rely on operator intervention.
- ISOLATETIME parameter: The default value of ISOLATETIME(0) allows other systems to take immediate action to isolate the failed system.
- DEACTTIME parameter: Specify a low value such as 0 or 10.

It is highly recommended that you enable the Automatic I/O Interface Reset Facility.

XCF can also automatically detect when a system in the Sysplex has been manually reset or re-IPLed when enabled and configured to use the system status detection (SSD) partitioning protocol. SSD exploits BCPii interfaces to use z Systems hardware services to discover the status of failed systems in a sysplex allowing for the bypassing of the failure detection interval, cleanup interval and the need for system fencing and manual operator intervention.

Setting up Tivoli System Automation

Depending on the system automation option you have chosen for your installation, you need to install and set up one or more Tivoli System Automation products.

Refer to the following topics to find required further information:

- Chapter 8, “Customizing Tivoli System Automation for z/OS,” on page 145 describes how to set up Tivoli System Automation.
- Chapter 9, “Customizing System Automation for Multiplatforms,” on page 185 describes how implement a highly available SAP system using System Automation for Multiplatforms.

SAP installation planning considerations

The contained subtopics inform you about required SAP licenses and recommendations for SAP logon groups.

When you install an SAP application server instance, you are prompted for the host name of the database server. Specify the host name associated with the static VIPA of the LPAR where the primary DB2 member runs to which this application

server should initially connect. An additional setup as described in “DB2 connection failover” on page 15 enables the failover of the application server instance to other DB2 members.

SAP licenses

SAP licenses depend on the hardware key of the server (CEC) on which the SAP messages server runs. For a high availability configuration, you should request and install an SAP license for each CEC that might possibly host the SAP Central Services.

The same is true in a heterogeneous environment if you run the SCS on another application server platform that is supported by the IBM SAP solution on z Systems and also by Tivoli System Automation for Multiplatforms.

You should install the SAP license before activating the SAP policy for z/OS. This is advisable, because the move groups for the SAP Central Services can start the Central Services on any CEC. The temporary license is installed only once for the CEC where the message server was running during installation of the SAP database instance. If SA z/OS moves the message server to another CEC afterward, an SAP error occurs in the SAP system during the license check.

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 the 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 can potentially run: :

```
saplicense -get
```

This provides you with all hardware keys for which you should request license keys from SAP.

The described standard SAP mechanism for retrieving hardware keys has the effect that even a capacity upgrade of your z Systems hardware might cause a change in the model number, which is part of the hardware key. Such a change means that you need to request and install a new SAP license in your SAP system.

SAP introduced an alternative mechanism that avoids a change of the hardware key after model-number changes of your z Systems hardware.

SAP Note 1367336 describes how to implement this mechanism that is especially recommended for high availability environments.

SAP logon groups

It is recommended that you define SAP logon groups. SAP 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 reconnecting to another SAP instance in case the SAP GUI connection or the instance itself becomes unavailable.

Use SAP transaction SMLG to set up logon groups that cover at least two application servers.

Note: You must use the virtual host name of the ABAP SCS as the message server host name if you run ABAP-only or dual-stack application servers. If you are running a highly available SAProuter with its own virtual host name, adapt the routing string in the logon group definition as well.

Message server logon group handling during restart

When the SAP message server is restarted, for example, as part of a maintenance operation, it usually takes considerable time until the message server has recovered its information about SAP logon groups. Users and RFC connections using these logon groups cannot log on during this time. By implementing *SAP Note 1787163: Message Server: save logon groups feature*, you can shorten this delay significantly. With SAP Kernels 7.4x, this feature is enabled by default and does not need to be switched on explicitly.

Chapter 7. Customizing SAP for high availability

The subtopics contained in this information unit describe the set-up of SAP to run as a high availability SAP solution.

The following topics are discussed:

- “General installation recommendations” on page 114
- “Setting up ABAP SCS, Java SCS, and ERS instances” on page 116: How to configure the ABAP and/or Java variants of SAP Central Services (SCS). Each SCS variant comprises the so-called standalone enqueue server. It also describes the configuration of the corresponding enqueue replication servers (ERS).
- “Installing SAP primary and additional application server instances” on page 128: How to install more than one application server instance to achieve redundancy for SAP application servers.
- “The SAP Web Dispatcher” on page 131: How to install the SAP Web Dispatcher, that connects the internet and the SAP system.
- “The Solution Manager Diagnostics Agent (SMDA)” on page 133: How to install and setup the SMDA instance, especially on z/OS with the so called *Agent on-the-fly* feature.
- “Preparing SAP on z/OS for automation” on page 135: How to enable IBM Tivoli System Automation for z/OS to manage SAP with the help of startup, monitoring and shutdown procedures.

You can also find detailed SAP documentation about how to install and use the SAP Central Services under:

<http://help.sap.com>

SAP NetWeaver 7.1 and newer releases are delivered with the following high availability-related changes, which were not delivered with SAP NetWeaver 7.0. These changes must be taken into account in the Tivoli System Automation *best practice* policy for SAP. The changes are:

- The **ZSCSinst** tool supports the installation of SAP ERS for ABAP and Java as individual SAP instances, including ERS instance profiles.
- There is now only one profile per instance that contains all the information, which was previously spread across two separate profiles, the instance profile and the instance start profile.

The Solution Manager Diagnostics Agent (SMDA) is a component of the SAP Solution Manager system landscape. For newer SAP NetWeaver releases, the SMDA is installed automatically together with the SAP application server instance. For installation details, see the *SAP Installation Guide* for your SAP application server platform.

With Software Provisioning Manager 1.0 SP09, you can install the SMDA instance on z/OS as a standalone engine with the **ZSCSinst** tool and set up the Solution Manager connection automatically during the installation (see “How to install and set up the SMDA instance on z/OS” on page 134).

From a high availability point of view, the function of SMDA is not seen as a single point of failure.

- For z/OS, the automation policy for the SMDA is described in Chapter 8, “Customizing Tivoli System Automation for z/OS,” on page 145.
- For AIX or Linux application server platforms, it can be automated using the SAP HA policy of Tivoli System Automation for Multiplatforms.

General installation recommendations

This topic provides important overall information for installing SAP Central Services for ABAP and Java.

SAP Central Services for ABAP, or Java, or both may be installed on z/OS UNIX System Services (z/OS UNIX) or on any other supported application server platform of the SAP on z Systems solution. They must be installed with an own virtual host name, which allows you to run them on different LPARs/machines.

For high availability reasons it is recommended that you run the SAP Central Services on z/OS UNIX and manage them through System Automation for z/OS. This recommendation is valid for both ABAP and Java Central Services.

In order to use the **SAPSRV add-on policy* with standard TCPIP-based replication, you must run the ERS as an own SAP instance under z/OS UNIX and install it with its own virtual host name, which allows you to run it on different LPARs/machines.

ZSCSinst is the installation tool delivered with the SAP Software Provisioning Manager (SWPM) and it should be used for all SAP NetWeaver installations on z/OS, even if you (re)install the SAP central services for older (7.0x) NetWeaver releases. *ZSCSinst* is a prerequisite for using the SAP HA Wizard.

Allocate the globally shared file systems needed by SAP on z/OS and export them via a highly available z/OS NFS server.

If you plan to run SAP Central Services on any of the following platforms, use Tivoli System Automation for Multiplatforms to automate them (see “Option 2: Automation by System Automation products with central components on AIX or Linux” on page 26):

- AIX
- Linux on z Systems
- Linux on System x

Installation sequence

The information in this sequence describes an SAP dual-stack (ABAP + Java) installation process. You can use a subset of this sequence for single-stack (ABAP or Java) installations.

About this task

The following sequence describes an SAP dual-stack (ABAP + Java) installation process, for example, for an SAP NetWeaver PI 7.1 system. For single-stack (ABAP or Java) installations, skip the installation parts for the stack that you are not using.

Note: For SAP NetWeaver 7.5 and higher, only single-stack installations are supported. With the release upgrade to SAP NetWeaver 7.5, a split from dual-stack to single-stack is mandatory.

Install at least one instance on each LPAR, on which you want to run the SAP Central Services. In a two-LPAR environment, that means, for example, to install the ASCS instance on LPAR1 and the ERS instance on LPAR2. That ensures that the SAP environment is correctly set up on each of the two LPARs.

Install a dual-stack SAP system in the following sequence:

Procedure

1. Install ASCS (the ASCS virtual host name must be active)
2. Install ERS for ASCS on a different LPAR (the ABAP ERS virtual host name must be active)
3. Restart ASCS to activate the enqueue replication function
4. Install SCS (the SCS virtual host name must be active)
5. Install ERS for SCS on a different LPAR (the Java ERS virtual host name must be active)
6. Restart SCS to activate the enqueue replication function
7. Install/load the SAP DB2 database
8. Install the primary application server (PAS)
9. Install one or more additional application servers (AAS)

Results

Alternatively you can install the system in a different order, that is, ASCS, SCS and then their ERSs. A prerequisite for the ERS installation is that its corresponding ASCS or SCS are installed.

Ensure that the corresponding VIPAs are active on the systems where the SAP instances are to be started. Start the SAP system manually in the following sequence:

1. Start ASCS
2. Start ABAP ERS on different system
3. Start SCS
4. Start Java ERS on different system
5. Start DB2
6. Start the primary application server (PAS)
7. Start one or more additional application servers (AAS)

Note:

- If you installed your application servers that use virtual host names, make sure that their VIPAs are active and the host names can be reached in your network before you start SAP.
- With SAP NetWeaver 7.5, the concept of an ABAP primary application server instance with instance name DVEBMGS<nn> are removed. All SAP ABAP instances are now installed with instance names D<nn> and are functionally equivalent.

Setting up ABAP SCS, Java SCS, and ERS instances

This information describes how to configure the ABAP and Java variants of SAP Central Services (SCS). Each SCS variant may contain an enqueue replication server (ERS) instance. Therefore, this topic also describes the configuration of the corresponding ERS instances. To install the mentioned components, follow the instructions provided in the contained subtopics. Start with implementing the prerequisites documented in this parent topic.

About this task

An SAP system running on IBM z Systems consists of a number of SAP instances which represent different portions of the complete SAP system. If this SAP system is configured for high availability, it contains the following instances:

- SAP application server instances, that run on AIX, Linux, or Windows:
 - ABAP-only instances
 - Java-only instances
 - ABAP and Java (dual-stack) instances
- SAP central services (SCS) instances, that run on z/OS UNIX System Services (z/OS UNIX) and are managed via IBM Tivoli System Automation for z/OS:
 - ABAP central service instance
 - JAVA central service instance
 - enqueue replication server (ABAP or Java) instance

Before you can start installing the SAP Central Services (SCS) under z/OS UNIX with the SAP installer tool *ZSCSinst*, you must complete certain prerequisites as documented in the following.

Note: The information is also relevant for the installation of the SCS under AIX or Linux on z Systems or Linux on System x.

Procedure

1. Set up a highly available network. You must use lowercase (virtual) host names. Refer to the sample TCPIP.PROFILE provided in “Network setup” on page 293. As the GLOBALTCPIPDATA statement in this sample is GLOBALTCPIPDATA(/etc/resolv.conf), the mapping in this case would have to be done in the file */etc/resolv.conf*.

Note: If you want to defer the network setup step, then you must, as a minimum, define the virtual host names used in the SAP installation to refer to already active IP addresses. Keep in mind that such a procedure means that you must reassign the correct IP addresses of the correct subnets afterwards. This in turn means that you must verify the functionality of the complete SAP system again.

2. Define and activate the dynamic and static VIPAs defined within your z/OS TCP profile (see “z/OS VIPA usage for the high availability solution for SAP” on page 58). Make sure that the VIPAs can be reached under their corresponding virtual host names. The dynamic VIPAs and their corresponding virtual host names are necessary in order to move applications (here NFS server, ASCS/SCS instances, and ERS instances) from one cluster node to another within a cluster. Each application and instance requires its own virtual host name so that it can be moved independently from another.

As a minimum you need:

- One dynamic VIPA for the highly available z/OS NFS server
- One dynamic VIPA for ASCS instance and another one for its ERS instance
- One dynamic VIPA for SCS instance and another one for its ERS instance
- n static VIPAs, where n is the number of LPARs which run a DB2 data sharing member for your SAP system. You may alternatively use n own dynamic VIPAs, one for each DB2 data sharing member. Then you must ensure that the dynamic VIPA is started together with the corresponding DB2 DS member on the correct LPAR. However, this alternative is not supported by the standard **DB2 add-on policy*, so that in this case you need to manually adapt the policy.

It is suggested that you define your virtual host names using the following naming conventions with lowercase virtual host names as required:

Used for	Static or dynamic VIPA	Sample virtual hostname	IP address
Highly Available z/OS NFS Server	dynamic VIPA	sapnfsv	
ASCS instance	dynamic VIPA	halascsv	
ABAP ERS instance	dynamic VIPA	halaersv	
SCS instance	dynamic VIPA	halscsv	
Java ERS instance	dynamic VIPA	haljersv	
(LPAR of) Data-sharing member n	Static VIPA	coh<n>vipa (COH<N> is the LPAR name)	

If you decide to run the SAP Web Dispatcher or the SAProuter, or both, under z/OS UNIX you need another dynamic VIPA for each.

You can activate a dynamic VIPA via a command from z/OS UNIX. For example if 10.101.5.194 is the IP address of halascsv, an authorized user can activate it using the following command:

```
moddvipa -p tcpip -c 10.101.5.194
```

3. Set up a high availability NFS server as described in Chapter 6, "Preparing a high availability SAP solution," on page 89
4. Define group and user IDs and add <sid>adm to the z/OS UNIX USERIDALIASTABLE, according to the *SAP Security Guide for IBM DB2 for z/OS*, and the *Planning Guide for SAP on IBM DB2 for z/OS*.
5. Download the latest version of *SAP Software Provisioning Manager* for z/OS from the **SAP Software Download Center**.
6. Set up WLM definitions according to the *Planning Guide for SAP on IBM DB2 for z/OS*.
7. Set up the file systems as described in Chapter 6, "Preparing a high availability SAP solution," on page 89.

Installing SAP Central Services and enqueue replication server on z/OS

All SAP components that run on z/OS UNIX System Services (z/OS UNIX) are installed with the SAP installation tool **ZSCSinst**. The **ZSCSinst** tool is implemented as a REXX script, which is called from the z/OS UNIX command line and can be used either interactively or using a response file that contains the installation parameters.

ZSCSinst is available as part of SAP's Software Provisioning Manager (SWPM). For details and invocation of **ZSCSinst**, refer to the *SAP Installation Guide* of the SAP component that you are planning to install.

Note: Whenever an SAP component is installed, the SAP installation tool either installs an SAP host agent or updates an existing version, if a newer version is available. In the second case, the system automation flag in SA z/OS should be set to NO for the SAP host agent on the CEC where you plan to run **ZSCSinst**. This is because **ZSCSinst** shuts down and restarts the SAP Host Agent after installation. This stop and start action comes into conflict with an active SA z/OS policy.

Installing the enqueue replication server as an ERS instance

Starting with SAP NetWeaver 7.1 and subsequent versions, SAP installs the enqueue replication server as its own instance.

If you are using the *EnqCF replication* mechanism (see Chapter 13, "Enqueue replication into a z Systems coupling facility," on page 277) you do not need to install an ERS instance.

Virtual host name for the ERS instance

A stand-alone ERS instance requires a virtual host name, which was not used in older SAP installations and therefore was often chosen to be the same as the virtual host name of its associated central services instance. The **SAPSRV add-on policy* requires that you install ERS with its own virtual host name. Therefore, you must use a different additional virtual host name, the ERS virtual host name. The advantages of defining the ERS instance with its own virtual host name are:

- It is more consistent to have a separate set of virtual host names for each ERS and its associated SCS or ASCS.
- A separate virtual host name for the ERS instance allows easier monitoring of the ERS or identification of the host on which it is running. Its VIPA is active on the host on which the ERS instance is running.

If you run a dual-stack SAP system, you need two additional virtual host names and corresponding VIPAs, which are used to make the ERS instance profile names independent from the real host names.

The VIPA / virtual host name is not used by the ERS itself because the ERS itself does not listen for client connect requests. But the VIPA must be active on the ERS LPAR such that the ERS `sapstartsrv` process can be reached from a remote host to start, stop, and monitor the ERS instance.

Installing Central Services and its enqueue replication server with ZSCSinst

Read all the required information on how to install Central Services and its enqueue replication server with **ZSCSinst**.

Before you start the installation under z/OS UNIX, make sure that the following variables are set for your installation user (UID = 0) are set:

```
_BPX_SHAREAS=NO
TZ=<your_current_timezone>
```

The following environment variables are inserted by **ZSCSinst** into the `.login` profile of the `<sid>adm` user:


```

| setenv _BPXK AUTOCVT ON
| setenv _TAG_REDIR_IN TXT
| set _TAG_REDIR_IN=TXT
| setenv _TAG_REDIR_OUT TXT
| set _TAG_REDIR_OUT=TXT
| setenv _TAG_REDIR_ERR TXT
| set _TAG_REDIR_ERR=TXT

```

Note: The setting `setenv _BPXK_AUTOCVT ON` is only required if you do not have *Enable Enhanced ASCII (Auto Conversion)* enabled at the LPAR level.

With the ZSCSinst tool, you can install the Central Services on z/OS UNIX either through an interactive command line dialog or by using predefined parameter files. The ZSCSinst tool provides sample parameter files for an ASCS, SCS and ERS installation, which you must adapt to your system definitions. The virtual host names must be entered as the value of the parameter VIRTUALHOST within those files.

In the examples with SID (SAPSID) HA1, the following dynamic VIPA names are used as VIRTUALHOST:

```

halascsv VIPA for the ABAP Central Services (ASCS)
halaersv VIPA for the ERS of ABAP Central Services
halscsv VIPA for the Java Central Services (SCS)
haljersv VIPA for the ERS of Java Central Services

```

You must activate the dynamic VIPAs manually on the LPAR on which you plan to do the installation, for example, by using the `moddvipa` z/OS UNIX command as described in “Setting up ABAP SCS, Java SCS, and ERS instances” on page 116.

Example invocation of ZSCSinst for an **SAP 740 SR2 ABAP Central Services Installation** (ASCS) using a parameter file:

```
./zscsinst 740SR2 ASCS /u/admin/REXX/ASCS.HA1
```

Example parameter file ASCS.HA1 for ABAP Central Services:

```

SYSTYPE=ASCS
HOSTNAME=halascsv
SAPSID=HA1
MOUNTDIR=/sapmnt
KERNELCD=/common/sapdvds/SAP_NW740/51046681_2/SAP_Kernel_7.40_z_OS_64bit_
        /DATA_UNITS/K_740_U_OS390_64
INSTANCENUMBER=00
MSPORT=3600
IMSPORT=3900
FQDN=boeblingen.de.ibm.com

```

Example invocation of ZSCSinst for the **SAP 740 SR2 Enqueue Replication Server (ERS) for ABAP Central Services**:

```
./zscsinst 740SR2 ERS /u/admin/REXX/ERSA.HA1
```

Example parameter file ERSA.HA1 for ERS of ABAP Central Services:

```

SYSTYPE=ERS
HOSTNAME=halaersv
SCSINSTANCEPROFILE=/sapmnt/HA1/profile/HA1_ASCS00_halascsv
INSTANCENUMBER=10
FQDN=boeblingen.de.ibm.com

```

Example invocation of ZSCSinst for a **SAP 740 SR2 Java Central Services Installation** (SCS):

```
./zscsinst 740SR2 SCS /u/admin/REXX/SCS.HA1
```

Example parameter file SCS.HA1 for Java Central Services:

```
SYSTYPE=SCS
HOSTNAME=ha1scsv
SAPSID=HA1
MOUNTDIR=/sapmnt
KERNELCD=/common/sapdvds/SAP_NW740/51046681_2/SAP_Kernel_7.40_z_OS_64bit_
/ DATA_UNITS/K_740_U_OS390_64
INSTANCENUMBER=01
IMSPORT=3901
FQDN=boeblingen.de.ibm.com
```

Example invocation of **ZSCSinst** for the **SAP 740 SR2 Enqueue Replication Server (ERS) for Java Central Services**:

```
./zscsinst 740SR2 ERS /u/admin/REXX/ERSJ.HA1
```

Example parameter file ERSJ.HA1 for ERS of Java Central Services:

```
SYSTYPE=ERS
HOSTNAME=ha1jersv
SCSINSTANCEPROFILE=/sapmnt/HA1/profile/HA1_SCS01_ha1scsv
INSTANCENUMBER=11
FQDN=boeblingen.de.ibm.com
```

The **SCSINSTANCEPROFILE** parameter contains the path and file name of the ASCS or the SCS instance profile for which the ERS must be installed.

After you have successfully installed the ABAP SAP Central Services or Java SCS or both, you must make additional manual modifications. Refer to the appropriate sections that follow for more information.

Additional mandatory manual modifications after ASCS installation

After installing ABAP SCS (ASCS) with a virtual host name, you must make some additional mandatory manual changes. There are also some optional changes that you can make if required.

Mandatory manual modifications

The following modifications make sure that the sapstartsrv restart mechanisms from SAP and the **SAPSRV add-on policy* work seamlessly together. This is necessary, because Tivoli System Automation relies on the SAP infrastructure to perform a restart of the message server (MS) in place. For details see *SAP note 768727: Process automatic restart functions in sapstart*.

Modifications when using the **SAPSRV add-on policy*:

- In the SCS or ASCS Instance Profile, the message server (MS) should be able to be restarted by sapstartsrv:
 - Change the line that contains the start commands for the MS:
from: **Start_Program_<xx>** to **Restart_Program_<xx>**

Modifications when using replication into the coupling facility (CF), that is, without ERS:

1. Change the SCS or ASCS Instance Profile such that the enqueue server (EN) can be restarted in place on the same LPAR:
 - Make sure that the start command for the EN specifies:
Restart_Program_<xx>

2. Since the ERS is not needed in this setup, you can uninstall the ERS instance by removing the ERS profile and its instance directory. If you want to keep the ERS installation, but want to avoid that the ERS instance is permanently shown in error in the SAP management console (SAP MC), you can alternatively rename the instance profile and the instance directory of the ERS and re-start the SAP MC.

Modifications when using the standard replication mechanism with ERS:

1. In the ERS Instance Profile:
 - set
Autostart = 0
 - change the start command for the ERS:
from: **Restart_Program_<xx>** to: **Start_Program_<xx>**
2. In the SCS or ASCS Instance Profile:
 - make sure the start command for EN specifies
Start_Program_<xx>

Adapting /etc/services:

If your /etc/services is not shared between LPARs, you must ensure that the port names used for ASCS are defined in /etc/services on all LPARs/hosts where it can run.

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

```
sapmsHA1 3600/tcp # SAP System Message Server Port
sapdp00 3200/tcp # SAP System Dispatcher Port
sapdp00s 4700/tcp # SAP System Dispatcher Security Port
```

Adaptions for ABAP SCS on AIX or Linux

The following steps are required for AIX and Linux, but are not necessary for z/OS UNIX.

The SAP installer creates the instance directories on the installation node. If you do not use a shared file system like NAS or GPFS™ / OCFS, then you must create the directories where the ABAP SCS (ASCS) instance and the enqueue replication server can run locally on each of the other nodes. This is the easiest way to have them available on all cluster nodes where the ASCS instance can run.

1. As <sid>adm on the installation node, enter the following commands:


```
cd /usr/sap/<sid>
tar -cvf ASCS00.tar ASCS00
scp ASCS00.tar <sid>adm@<other_cluster_node>:/usr/sap/<sid>
```
2. As <sid>adm, on each other node where the ASCS can potentially run, enter the following commands:


```
cd /usr/sap/<sid>
tar -xvf ASCS00.tar ASCS00
rm ASCS00.tar
```

If you run ASCS under z/OS UNIX, do not set the following parameter in the instance profile of ASCS:

enque/backup_file

This means the default value is used, which is `/usr/sap/<sid>/ASCS<instance no.>/log/ENQBCK`. Because it is recommended that you use a shared zFS file system under z/OS UNIX, the file is automatically accessible from all SYSPLEX LPARs.

For Linux and AIX: To share the file within the Linux or AIX cluster you **must** set the parameter `enqueue/backup_file = $(DIR_GLOBAL)/ENQBCK`. It is then shared via the exported NFS `/sapmnt/<sid>/global` directory.

Verifying ABAP SCS with enqueue replication

When you complete the steps that are described in the previous sections, you must verify that your SAP central services installation runs without problems.

Starting and stopping the ASCS

Verify that you can start and stop the ASCS services manually:

1. Activate the VIPA under z/OS UNIX System Services (z/OS UNIX) or the IP alias under Linux. Under z/OS UNIX, use the `moddvipa` command as root user:
`moddvipa -p tcpip -c 10.101.5.194` (if this is the IP address of `halascsv`)

Under Linux, use the `ifconfig` command as root user:

```
ifconfig eth0:1 <virt_host_IP> broadcast <broadcast ip>
                                netmask <netmask> mtu 1500 up
```

Note: Enter the command `ifconfig -a` to check whether the alias `eth0:1` is already in use.

2. Manually start the ASCS00 instance as `<sid>adm`:
`startsap r3 ASCS00 halascsv`
3. Verify that all processes are running correctly with `sapcontrol`:
`sapcontrol -nr 00 -function GetProcessList`

The output must show that all processes have a **GREEN** status as shown in this example output:

```
12.11.2013 07:50:22
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
msg_server, MessageServer, GREEN, Running, 2013 11 11 10:56:59, 20:53:23, 67175064
enserver, EnqueueServer, GREEN, Running, 2013 11 11 10:56:59, 20:53:23, 33620546
```

4. If you are not using *EnqCF replication*, manually start the enqueue replication server instance ERS10 as `<sid>adm` on a different LPAR after you activated the ERS VIPA there:
`startsap r3 ERS10 halaersv`

Verify that the replication process is running correctly with `sapcontrol`:

```
sapcontrol -nr 10 -function GetProcessList
```

The output must show that the process has a GREEN status as shown in this example output:

```
2.11.2013 07:55:02
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
enrepsrver, EnqueueReplicator, GREEN, Running, 2013 11 12 06:51:58, 1:03:04,
...
```

Verification of the replication mechanism

Start an SAP application server and use SAP transaction SM12 to generate test entries in the enqueue table to verify manually that the SAP system is running with replication enabled. See “Preparing for the test” on page 201 for further information. It is only possible to generate entries and display them when the enqueue server of ASCS is running, and not when it is stopped.

1. To check whether replication is active (either via the replication server or by using *EnqCF replication*), use:

```
ensmon pf=<profile> -H <hostname>
```

For example, enter:

```
ensmon pf=/usr/sap/HA1/SYS/profile/HA1_ASCS00_ha1ascsv
```

2. From the main menu, select **1: Dummy request**. The dummy request must run successfully.
3. Select **2: Get replication information**. This returns:

```
Replication is enabled in server, repl. server is connected
Replication is active
```

and also displays statistics.

Note: The output shows repl. server is connected even when using *EnqCF replication*, where there is no separate replication server running.

4. To display the generated test entries, run:
5. Kill the enqueue server and message server manually.
6. Move the VIPA to the system where the replication server is started. Start the ASCS instance there. The replication server must stop itself after the enqueue table has been rebuilt.
7. Run again:

```
enqt pf=/usr/sap/HA1/SYS/profile/HA1_ASCS00_ha1ascsv 20
```

If it displays the same output as before you killed the enqueue server, you have verified that replication works.

Additional mandatory manual modifications after Java SCS installation

After you installed the Java SCS with virtual host name, you must make some additional and mandatory manual changes.

The SCS and Java ERS instance profiles need to be changed similar to the recommendations for ABAP SCS and ABAP ERS (see “Additional mandatory manual modifications after ASCS installation” on page 120).

If your `/etc/services` is different for each LPAR/host, make sure that the port names that are used for the installed SAP Central Services are defined in `/etc/services` on all LPARs/hosts where it can run.

For Java SCS, similar adaptations of the `/etc/services` files might be required as described in paragraph *Adapting /etc/services* in “Additional mandatory manual modifications after ASCS installation” on page 120.

If Java SCS runs under Linux or AIX, you must add the following entry in the instance profile of the Java SCS:

```
enque/backup_file = $(DIR_GLOBAL)/ENQBCK_JAVA
```

The enqueue backup file is then shared via the NFS exported `/sapmnt/<SID>/global` directory. This sample assumes that you run the SCS in a dual-stack installation. Then, you must also add such an entry to the ASCS instance profile. To distinguish between the ASCS and Java SCS enqueue backup file, you must define two different names.

To ensure proper functions of SAP utilities such as `ensmon` or `enqt` in a dual-stack installation (ABAP and Java stack) check the Java SCS instance profile. It must contain the parameter `enque/serverhost` set to the Java virtual host name. If the parameter is missing in the SCS profile and you are using these utilities, they fail because they use the `enque/serverhost` from the default profile `DEFAULT.PFL`, which is set to the ABAP SCS virtual host name.

Adaptions for Java SCS on AIX or Linux

The steps that duplicate the Java SCS installation directories to other nodes are required for AIX and Linux, but are not necessary for z/OS UNIX. For details see “Adaptions for ABAP SCS on AIX or Linux ” on page 121.

Verifying Java SCS with enqueue replication

You must verify that your dual-stack system runs without problems with the virtual host name and the replication server. You can start and stop the SCS services manually. This topic presents the steps and commands how to achieve this.

1. First, you must activate the VIPA under z/OS UNIX System Services or the IP alias under Linux. Under z/OS UNIX, use the `moddvipa` command as root user:

```
moddvipa -p tcpip -c 10.101.5.195 (if this is the IP address of halscsv)
```

Under Linux, use `ifconfig` command as root user:

```
ifconfig eth0:2 <virt_host_IP> broadcast <broadcast_ip>
                                netmask <netmask> mtu 1500 up
```

Note: Enter the command `ifconfig -a` to check whether the alias `eth0:2` is already in use.

2. Manually start the SCS01 instance as `<sid>adm`:

```
startsap r3 SCS01 halscsv
```

Verify that all processes are running correctly with `sapcontrol`:

```
sapcontrol -nr 01 -function GetProcessList
```

The output must show that all processes have a GREEN status, similar to the output example shown in “Starting and stopping the ASCS” on page 122.

3. Manually start the enqueue replication server as `<sid>adm` on another machine after you have activated the Java ERS VIPA there:

```
startsap r3 ERS11 haljersv
```

Verify that the replication process is running correctly with `sapcontrol`:

```
sapcontrol -nr 11 -function GetProcessList
```

The output must show that all processes have a GREEN status, similar to the output example shown in “Starting and stopping the ASCS” on page 122.

Verify manually afterward that the SAP SCS is running correctly. Use the utility **enqt** that SAP provides. Run as `<sid>adm`:

```
enqt pf=/usr/sap/HA1/SYS/profile/HA1_SCS01_ha1scsv 97
```

The output is similar to:

```
---REQ-----  
EnqId:          EnqTabCreaTime/RandomNumber   = 06.09.2015 00:06:19 1125957979 / 8563  
ReqOrd at Srv:  TimeInSecs/ReqNumberThisSec   = 09.09.2015 13:45:43 1126266343 / 1  
-----
```

where `EnqId` is the unique identifier of the enqueue server and its enqueue table.

In addition, run:

```
enqt pf=/usr/sap/HA1/SYS/profile/HA1_SCS01_ha1scsv 20
```

The output is similar to:

```
J2E <interna $service.e X ejb/CreateEmptyImageBean  
J2E <interna $service.e X ejb/FinishImageBean  
J2E <interna $service.j X  
Number of selected entries: 3
```

This shows the current enqueue table entries.

Use the `ensmon` utility to check whether the replication server has successfully connected to the Java stand-alone enqueue server. For example, run:

```
ensmon pf=/usr/sap/HA1/SYS/profile/HA1_SCS01_ha1scsv
```

From the main menu select:

- **1: Dummy request** . The dummy request is run successfully.
- **2: Get replication information**. This returns the information that Replication is enabled in server, repl. server is connected Replication is active and displays the statistics lists.

Stop the enqueue server and the message server manually. Move the VIPA to the system, where the replication server is started and start the SCS instance there. The replication server must stop itself after the enqueue table has been rebuilt.

Run:

```
enqt pf=/usr/sap/HA1/SYS/profile/HA1_SCS01_ha1scsv 20
```

again. If it displays the same output as it did before you stopped the enqueue server, you verified that replication works.

Note: As described in “Adapt the Java property `ms.reconnect.timeout`” on page 131, a Java application server stops and restarts if it can reconnect to the Java message server within the time that is defined by property `ms.reconnect.timeout`. If your manual failover takes longer than your timeout setting, adapt that property for manual failover tests.

SAP profile parameters

The tables in this information unit list and describe the profile parameters that are related to ASCS/SCS.

Table 11. Enqueue client parameters relevant for the high availability solution

Parameter	Description	Default value	Recommended value
ms/persist_lg_info	Information on logon groups can be stored regularly and during stop operations of the message server. It can be restored after its restart, so that users and RFC connections using these logon groups can directly log on after the message server restart.	Kernel 7.2x: OFF Kernel 7.4x: ON	SAP Kernel 7.2x: ON SAP Kernel 7.4x: no need to set it, since the default value is: ON This parameter setting assumes that the recommended file system setup is done on the SAP Central Services instance directories on a shared zFS file system (see "Setting up zFS file systems" on page 97). The message server creates the file \$(DIR_HOME)/<SID>_msg_server_adtl_storage and updates it on a regular basis. For more details see <i>SAP Note 1787163</i> .
enqueue/serverhost	Host 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)
enqueue/dequeue_wait_answer	Indicates whether a dequeue request waits for the acknowledgment. If the default value (FALSE) is used, obsolete locks might remain in the enqueue table on failover and must be removed manually. If TRUE is specified, the reported enqueue time of all transactions increases slightly.	FALSE	TRUE
enqueue/sync_dequeall	When an SAP transaction is ended, the system uses a final DequeueAll() to clean up the system resources. This DequeueAll() is an asynchronous request, where a chance remains that the system assumes its DequeueAll() was sent, although it may never arrive at the standalone enqueue server, for example, if a network problem occurs. The related locks are retained in such a case, which does not result in inconsistencies, but could block other transactions.	0	1

Table 11. Enqueue client parameters relevant for the high availability solution (continued)

Parameter	Description	Default value	Recommended value
enqueue/server/repl/async	<p>Indicates whether replication of enqueue locks should be done in an asynchronous mode. If this parameter is set to TRUE, this may cause a loss of enqueue locks in case the enqueue server fails before the locks were sent - asynchronously - to the replication server. (see <i>SAP Note 2036171: Missing locks following failover between enqueue server and replicated enqueue server</i>).</p> <p>For high availability systems, the parameters must be set to FALSE.</p>	<p>Kernel 7.20: TRUE Kernel 7.21 and later: FALSE Kernel 7.4x: FALSE</p>	<p>SAP kernel 7.20: explicitly set to FALSE SAP kernel 7.21 or higher: no need to explicitly set the parameter, since the default value is OK.</p>
enqueue/con_retries	<p>Specifies the maximum number of times the enqueue client tries to establish a connection to the enqueue server.</p> <p>If you multiply the parameters enqueue/con_retries and enqueue/con_timeout (which has a default value of 5 seconds), the result is the maximum time an enqueue or dequeue request is retried. If you keep the default values, this means that an error is triggered after 300 seconds.</p> <p>Set this parameter to a value, so that the calculated result from the mentioned parameters is larger than the typical time for an ASCS or SCS instance failover. This depends largely on your operating system and your cluster solution. If a failover can take longer, you should increase the value of these parameters.</p> <p>With the settings recommended in this document, the time is ten minutes.</p>	60	120

Table 12. Enqueue server parameters relevant for the high availability solution

Parameter	Description	Default value	Recommended value
enqueue/server/replication	Enables replication.	false	true
enqueue/encni/repl_port	<p>Port number which the enqueue server opens for replication.</p> <p>Use this parameter if the replication port of the enqueue server cannot be set to the default port 5XX16 (where XX is the instance number). It should then be set either in DEFAULT.PFL, or in the profiles of the enqueue server and the replication server</p>	default port 5XX16 where XX is the instance number.	<p>5XX16</p> <p>If possible, do not specify this parameter. The default port 5XX16 is used in this case.</p>

Table 12. Enqueue server parameters relevant for the high availability solution (continued)

Parameter	Description	Default value	Recommended value
enqueue/server/threadcount	Number of I/O threads in the enqueue server.	1	1
enqueue/backup_file	<p>Specifies the complete path to the enqueue backup file. This is used to save the locks, which are transferred to the update, in case the enqueue server is intentionally stopped (including a stop of the replication server). If the enqueue server is restarted it reads the locks from this file again, so that updates which have not been processed can be processed. If a shared file system is used, such as zFS under z/OS UNIX, the default value is satisfactory.</p> <p>Note: If ASCS and/or SCS runs under Linux or AIX and no shared file system is used for the instance directory, it must be set to: enqueue/backup_file = \$(DIR_GLOBAL)/ENQBCK and enqueue/backup_file = \$(DIR_GLOBAL)/ENQBCK_JAVA</p> <p>It is then shared via the NFS exported /sapmnt/<SID>/global directory.</p> <p>The sample here assumes that you run the SCS in a dual-stack installation. Therefore you must also add such an entry for ABAP to the ASCS profile. To distinguish between ASCS and SCS enqueue backup file, you must define two different names.</p>	/usr/sap/<SID>/ ASCS<instance number>/log/ENQBCK or /usr/sap/<SID>/ SCS<instance number>/log/ENQBCK	

Installing SAP primary and additional application server instances

You can usually achieve redundancy for SAP application servers by installing more than one application server instance, which is supported by the NetWeaver architecture. The SAP high availability policy that is provided by Tivoli System Automation for Multiplatforms defines SAP application server resources as fixed resources that are not moved to another node.

From a high availability point of view, it is sufficient to install an application server using the physical host name. On the other hand, the SAP installer supports installation of an application server instance with its own virtual host name. Exploiting this feature makes it easier to move an instance from one physical machine to another in case this should be necessary, for example, when you migrate to newer and faster hardware. When installing SAP application servers using virtual host names, you should use these virtual host names in the client affinity definition in **db2dsdriver.cfg** as well.

To exploit the SAP high availability policy that is provided by Tivoli System Automation for Multiplatforms, and to get the flexibility provided by SAP's virtualization option, it is recommended that you install every application server with its own virtual host name, provided that the home directory of the <sid>adm is local on each machine. System Automation for Multiplatforms defines and manages the application server as a fixed resource regardless of whether its host name is virtual or physical. Do not use System Automation for Multiplatforms to

move such an application server from one physical host to a different one. This is performed as a manual task outside of automation after the cluster has been changed to include the new host.

Installing the SAP primary application server instance

The following describes an installation under AIX. Using Linux on z Systems or Linux on System x is also possible and supported.

For a Windows application server installation, see the SAP SCN article: SAP HA Installations on z/OS and Windows application servers.

As previously mentioned, it is recommended to install an application server with its own virtual host name. Under AIX, define a virtual interface (vi<x> interface) and under Linux, define a dummy θ device with the corresponding IP. The virtual host name and the corresponding IP must be defined either in the DNS or locally in /etc/hosts on all servers of the SAP system. The IP must be active before the installation starts.

To install SAP application servers (primary as additional application server) with a virtual host name, call the sapinst command with virtual host name parameter:

```
./sapinst SAPINST_USE_HOSTNAME=p570coh1v
```

Note: Make sure that System Automation under z/OS is not controlling the Java SCS during the primary application server installation. This is because SAPinst tries to start and stop the Java SCS during the installation procedure. This creates a problem if the Java SCS is already managed and kept highly available by System Automation at that time, and it can cause the primary application server installation to fail.

Installing additional SAP application server instances (Java-only)

Read this information if you require additional Java-only SAP AS instances being installed.

During the installation of additional Java application server instances, it is recommended that you reuse the existing Java database connection that was defined when the primary application server instance was installed.

In a subsequent step, set up the wanted failover configuration and workload distribution for all Java application servers:

- For workload distribution, modify the config.xml file, so that different Java application servers have primary database connections to different DB2 members.
- For failover configuration, define the failover sequence for each application server.

See the *Database Administration Guide: SAP on IBM DB2 for z/OS* for details and a sample config.xml file.

Performing SAP post installation steps for high availability

In addition to the post-installations steps outlined in the *SAP Installation Guide*, using a virtual host name requires you to perform platform-specific manual steps to ensure correct DB2 connection failover for ABAP work processes to a standby DB2 data sharing member.

Update the db2dsdriver.cfg configuration file

When using **CLI failover** and virtual host names for your SAP application servers, you need to adapt the connection profile `db2dsdriver.cfg`. The SAP installation inserts only standard host names and only one `<affinitylist>` entry and only one DB2 member into the CLI failover configuration file `db2dsdriver.cfg`. With this configuration, a database connection failover is not possible.

You must therefore adapt the `db2dsdriver.cfg` file as follows:

- replace the application server host names with the virtual host names
- insert additional data sharing members in the `<alternateserverlist>` section
- define additional entries in the `<affinitylist>` section and define failover sequences in the **serverorder** parameters
- assign the affinity lists to your application servers virtual host names as appropriate.

These changes can be made using the SAP provided *Failover Configuration Tool* in SAP transaction DBACOCKPIT. The tool ensures the syntactical correctness of the changes and can activate the new **CLI failover** configuration.

If you choose to adapt the file on the operating system level, you should validate your changes using the *validate* option of the **db2cli** utility that is part of the DB2 CLI driver. After successful validation you need to activate the changes.

Install SAP application server on different physical machines and define redundant services

For availability reasons you should have at least two application server instances installed:

1. a primary application server (PAS)
2. an additional application server (AAS) on another physical machine

The profile parameters of the additional application server instances must be configured manually so that all SAP ABAP services that run on the PAS installation are also on it. These are:

- Batch service
- Update/Update 2 service
- Spool service

Through this setup, all non-unique SAP services run on each of the application servers and therefore no longer constitute single points of failure (SPOF).

To ensure redundancy from the SAP user perspective you must configure the SAP logon groups.

Adapt the Java property `ms.reconnect.timeout`

By default, a Java application server stops and restarts if it can reconnect to the Java message server within three minutes. Three minutes is the default value of the property `ms.reconnect.timeout`. For more details, see the SAP online help library: <http://help.sap.com>

or see *SAP Note 1121900: Message server reconnect parameter optimization - AS Java 7.1*. This SAP Note also applies to Java application servers 7.2 and higher.

In order to not restart the Java application server in case of an LPAR loss, you should use a minimum value of 540000 (milliseconds) for `ms.reconnect.timeout`, which is nine minutes if you run your z/OS with shared CPs. Use a minimum of 300000 (five minutes) for dedicated CPs. These values take into account two times the default time of z/OS spin loop detection and grant time for SA z/OS to restart z/OS UNIX resources after an LPAR is lost.

Modify SAP Profiles

Enter the following into `DEFAULT.PFL` or into each application server instance profile:

```
enqueue/con_retries = 120
enqueue/deque_wait_answer = TRUE
enqueue/sync_dequeueall = 1
```

Starting with SAP 7.10 it is no longer necessary to explicitly set the profile parameter for the enqueue replication port:

```
enqueue/encni/repl_port
```

Use this parameter only if the replication port of the enqueue server cannot be set to the default port 5XX16 (where XX is the instance number).

The SAP Web Dispatcher

The SAP Web Dispatcher is required to connect the internet and the SAP system. Read this topic for installation information.

The SAP Web Dispatcher receives requests from the internet (http/https) and forwards the requests to an ABAP server or Java application server. The incoming requests are distributed among the application servers according to their capacity weighting (the amount of dialog work processes for application server ABAP and the amount of server processes for application server Java).

The SAP Web Dispatcher version 7.20 (or higher) must be installed with its own SAP system identifier (SID). This is because it can connect to multiple SAP Systems. An installation with a SID of its own makes sure that the SAP kernel level of the SAP Web Dispatcher remains independent of the kernel levels of the SAP systems to which it connects.

Installing the SAP Web Dispatcher

To install the SAP Web Dispatcher on z/OS UNIX System Services (z/OS UNIX), you must use the **ZSCSinst** tool that is part of SAP's Software Provisioning Manager (SWPM). For details and invocation of **ZSCSinst**, refer to the *SAP Installation Guide*.

To make the SAP Web Dispatcher highly available, you must install this SAP component with a virtual host name. Here is an example of a parameter file for **ZSCSinst** (you must enter the virtual host name as the value of the parameter **HOSTNAME**):

```
SYSTYPE=WD
HOSTNAME=sapwd1wdv
SAPSID=WD1
MOUNTDIR=/sapmnt
KERNELCD=/common/sapdvds/SAP_NW740/51046681_2/SAP_Kernel_7.40_z_OS_64bit_j
        /DATA_UNITS/K_740_U_OS390_64
INSTANCENUMBER=77
WMSHOST=ha3scsv
WMSPORT=8130
WDHTTP=8177
```

ZSCSinst invocation for an SAP 740 SR2 Web Dispatcher installation with **ZSCSinst** configuration file **WD.WD1**:

```
./zscsinst 740SR2 WD /u/wd1adm/SWPM/WD.WD1
```

Postinstallation steps

- It is mandatory to follow the postinstallation steps as described in the **ZSCSinst** installation documentation.
- Make sure that you use the minimum version of the Web Dispatcher executables including the **wdispmon** executable. See Table 6 on page 90 for the exact minimum levels and how to determine them.
- For installations with SAP kernel 7.20 or higher, configure the SAP Web Dispatcher to manage more than one SAP system. For details about the multi-system capability of the SAP Web Dispatcher, refer to the SAP online help and search for topic *SAP Web Dispatcher for Multiple Systems*.
- For details about the architecture of the SAP Web Dispatcher and its high availability capabilities, search in the SAP online help for topic *High Availability of the SAP Web Dispatcher*. The SA z/OS **SAPSRV add-on policy* implements the option *High Availability of SAP Web Dispatcher with External HA Software* that is mentioned in the SAP documentation.

The recommendation is not to use the restart capability (via profile parameter **Restart_Program_XX**) that is mentioned on this web page. This is because the **SAPSRV add-on policy* needs to restart the health check function **wdispmon** whenever the Web Dispatcher is restarted (see “SAP Web Dispatcher health checking” on page 133).

SA z/OS policy for the SAP Web Dispatcher

The **SAPSRV add-on policy* of SA z/OS 3.5, APAR level OA46166 or higher, contains improved resource definitions for the SAP Web Dispatcher.

- The SAP Web Dispatcher policy is also based on the standard SAP infrastructure and the SAP start service **sapstartsrv** to control SAP instances. So a Web Dispatcher instance is controlled via the **sapstartsrv** framework.
- In addition to standard system automation mechanisms (Address space monitoring and Process monitoring) to verify that the SAP Web Dispatcher is running, the policy implements and uses the SAP Web Dispatcher health checker mechanism. This mechanism can detect application problems within the SAP Web Dispatcher itself, and initiates a failover if necessary.
- The health check is implemented in the **SAPSRV add-on policy* with a new Web Dispatcher monitoring resource (MTR). This monitoring resource needs a REXX program, which is used to determine the correct health status, either **NORMAL** or **CRITICAL**. The REXX program is called **SAPWDMTR**. This program is

delivered with SA z/OS 3.5 and APAR level OA46166 and higher as part of the `ING_sap.tar` file in the z/OS UNIX directory `/usr/lpp/ing/SAP`. Add the **SAPWDMTR** REXX program to a user-defined data set within the NetView command list, for example `USER.SINGNREX`. These data sets are specified by the `DSICLD DD` statement of NetView.

SAP Web Dispatcher health checking

A disadvantage of the traditional address space and process monitoring is that it does not receive information if the SAP Web Dispatcher process hangs in a state where it no longer services incoming requests.

In order to handle such a situation, the **SAPSRV add-on policy* uses the SAP Web Dispatcher monitoring function `wdispmon`.

Note: *SAP Note 1601565* lists the minimum required SAP Web Dispatcher 7.20 level.

SA z/OS 3.5 APAR OA46166 additionally introduces an enhanced SAP Web Dispatcher automation policy based on the `sapstartsrv` framework. It uses an enhanced version of:

- script `start_cs`
- script `checkwd`

Both scripts are part of the `ING_sap.tar` file, which is shipped with SA z/OS in the z/OS UNIX directory `/usr/lpp/ing/SAP`.

Extract the tar file into the home directory of the SAP administrator user ID such that the SAP Web Dispatcher resource can use them:

```
/u/<sid>adm
```

The first script `start_cs` starts the SAP Web Dispatcher instance. It is used in the `STARTUP` section of the startup processing for the SAP Web Dispatcher `sapstart` resource (`SAP<SID>WD_ST`) in the changed SAP policy.

The second script `checkwd` implements a monitoring procedure for the SAP Web Dispatcher that relies on the SAP `wdispmon` function. This script is used in the `POSTSTART` section of the start-up processing for the SAP Web Dispatcher `sapstart` resource (`SAP<SID>WD_ST`) by the changed SAP policy.

For a more detailed description of the scripts, see Chapter 14, "Reference of the z/OS high availability scripts," on page 285.

The Solution Manager Diagnostics Agent (SMDA)

Read the contained information about the role of the Solution Manager Diagnostics Agent (SMDA) in an SAP high availability environment and how to install and set up an SMDA instance.

The Solution Manager Diagnostics Agent (SMDA) is a component of the SAP Solution Manager system landscape, which runs on the managed systems and connects to the SAP Solution Manager. It gathers information from the managed system and reports it to the SAP Solution Manager system.

Installing the SMDA instance

Follow the provided information to perform the installation of the SMDA instance for a high availability SAP environment.

The SMDA instance can be installed on several servers:

1. On non-z/OS application server platforms and for newer SAP Netweaver releases, it can be automatically installed together with the application server instance or manually afterward as standalone engine. For detailed information, see the *SAP Installation Guide* for your application server platform.
2. On z/OS UNIX, install one SMDA instance for each z/OS LPAR, which is involved in your high availability environment. For detailed information, see the *Installation and Setup Guide - Diagnostics Agent: z/OS*. Make sure that you have the latest version of this document, which can be downloaded from the SAP Support Portal.

If you run the SMDA on z/OS in the context of a high availability system landscape you have to take the following into account:

1. Terminology from SAP for <hostname>, which> is also described in this Wiki on the SAP Community Network:
Diagnostics Agent and HA Support
2. You must use the *Agents on-the-fly* feature. The concept of this feature is explained in the Wiki that is mentioned in step 1.
3. In the context of the *Agents on-the-fly* feature, it is not allowed to use the same SID with multiple instances (like 98, 97). You must select different SIDs like *DA1*, *DA2*, or *DA3* on the different hosts and the instance number can be the same or a different one.

How to install and set up the SMDA instance on z/OS

SMDA installation and setup on z/OS is described in this topic with the help of an example installation. Note that the shown example is for z/OS with SWPM 1.0 SP09.

For details, see the SAP Community Network:

- Diagnostics Agents
- Diagnostics Agent Maintenance Procedures
- Diagnostics Agent and HA Support

Prerequisites for SMDA on z/OS

Implement the following SAP Notes:

- *SAP Note 1870733: DB2-z/OS:CCMS: Database outside discovery enablement* to apply fixes to ensure correct SAP Host Agent behavior on z/OS
- *SAP Note 1881267: DB2-z/OS: SAP Host Agent problem on z/OS* to set up z/OS to enable automatic database detection by the SAP Host Agent (sapdbctrl)
- *SAP Note 1887279: DB2-z/OS: sapdbctrl: Prerequisites and Configuration* to enable the Solution Manager (ABAP-part) for database outside discovery.

Example installation and setup of an SMDA instance on z/OS

The following example describes the SMDA installation and setup on z/OS with SWPM 1.0 SP09.

The installation tool **ZSCSinst** is contained in the SAP installer SWPM for z/OS. You can run the script in prompt mode or you can create an input parameter file by copying the SMDA.DEFAULT skeleton to a new file.

You can complete this new file with your specific input parameter values. The parameters for an SMDA instance installation are explained in the *Installation and Setup Guide - Diagnostics Agent: z/OS*.

Important: For parameter **HOSTNAME**, you must choose the output of the **hostname** command, which should be the virtual host address (VIPA) in the high availability context.

ZSCSinst input parameter file sample:

```
SYSTYPE=SMDA
HOSTNAME=coh2vipa
SAPSID=DA2
KERNELCD=/common/pewa/SMDA/Kernel_742_0S390
INSTANCENUMBER=92
SLDYESNO=NO
SLDHTTPHOST=
SLDHTTPPORT=
SMDSLCONYESNO=YES
SMDP4TYPE=1
SMDP4HOST=solman.boeblingen.de.ibm.com
SMDP4PORT=8101
JAVADIR=/usr/lpp/java/J7.0_64
```

With SWPM 1.0 SP09 or higher, you have the possibility to automatically set up the connection to the Solution Manager during the installation with the **ZSCSinst** tool so that you no longer need to call the `smsetup` script manually after the installation.

After the successful installation, you can start your new SMDA instance on z/OS with your `<sid>adm` user ID as described in the *Installation and Setup Guide - Diagnostics Agent: z/OS*. The `<sid>adm` user ID stands for the Diagnostics Agent administrator who is authorized to install and manage the SMDA.

Deleting an SMDA instance

Read the contained information on how to delete an SMDA instance.

Currently, you must delete your SMDA instance on z/OS manually. See *SAP Note 1259982 - UNIX:Deleting an SAP System Based on NW 7.1 and Higher* and follow the instructions under topic *Deleting a Diagnostics Agent Manually*

Preparing SAP on z/OS for automation

The subtopics of this information unit describe the startup, monitoring and shutdown procedures that enable IBM Tivoli System Automation for z/OS to manage SAP.

These startup, monitoring and shutdown procedures are implemented in shell scripts that are installed in addition to the standard scripts installed by the SAP installation utility. The SAP standard scripts are not modified.

The scripts also write messages to the system console, thereby triggering immediate Tivoli System Automation for z/OS actions.

For a comprehensive list of scripts and other key files, see Chapter 14, “Reference of the z/OS high availability scripts,” on page 285.

Preparing the C-shell and logon profiles

System Automation for z/OS needs to execute z/OS UNIX commands on behalf of the SAP administrator to be able to manage SAP. Read this information on how to start the required scripts and commands.

Tivoli System Automation invokes these UNIX commands by starting the user's default shell and naming the shell script that is to be run (for example, `/bin/tcsh -c '<command>'`). The C-shell should be the default shell for the SAP administrator ID on z/OS.

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 `$HOME/.login` are *not* processed. This is the case when programs are invoked via BPXBATCH in a started task, or via the Tivoli System Automation command `INGUSS`. Therefore, make sure that all relevant settings that are needed for the startup of the SAP system are in the profiles `/etc/csh.cshrc` and `$HOME/.cshrc`. In particular, make sure that the default `umask` setting for non-login shells matches the setting for login shells.

If the environment variable `$HOME` is not yet set to its corresponding home directory for user `<sid>adm`, you need to set it now. For example, `setenv HOME /u/haladm` would set `$HOME` for SAP user `haladm`. It can be done for example in `$HOME/.cshrc`.

ABAP SAP Central Services (ASCS)

Read this information to learn about the purpose of the ABAP SAP Central Services and how to manage them with the help of scripts.

The ASCS consist of the following unique SAP resources that share the same instance profile and instance directory:

- ABAP enqueue server
- ABAP message server

In order to allow transparent failover of the ASCS to another system, the enqueue server must - during its restart - have access to the enqueue replication information in order to be able to reconstruct the enqueue table.

For traditional enqueue replication, a transparent failover of the ASCS to another system is only possible if the enqueue server is restarted on the system where the enqueue replication server was running, and if replication was active at the time the failover is triggered. For the *EnqCF replication* mechanism (see Chapter 13, “Enqueue replication into a z Systems coupling facility,” on page 277), a transparent failover of the ASCS to another system is possible on any LPAR which has access to the CF structures in which the replication information was saved.

The SAP process `sapstart` is the parent process of all ASCS resources, in particular the enqueue and message server. For fast failure detection and recovery, these

processes are monitored individually by SA z/OS. The **SAPSRV add-on policy* uses the `start_cs` shell script to start the instance for this purpose.

The `start_cs` script in turn needs the `sapstartsrv` infrastructure of the ASCS instance. Starting and stopping the `sapstartsrv` service is done in the **SAPSRV add-on policy* using the script `start_sapsrv`. See Chapter 14, “Reference of the z/OS high availability scripts,” on page 285 for a detailed description.

Scripts to start an ASCS instance are:

1. The `start_sapsrv` script starts the `sapstartsrv` service of the ASCS instance.
2. The `start_cs` script tells the `sapstartsrv` service to start the ASCS:
 - a. the `sapstartsrv` service starts the `sapstart` master process of the ASCS instance.
 - b. the `sapstart` master process finally starts the individual resources of the ASCS instance, namely the enqueue and message server.

Java Central Services (SCS)

Read this information to learn about the purpose of the Java Central Services (SCS) and how to manage them with the help of scripts.

Java SCS consist of the following unique SAP resources that share the same instance profile and instance directory:

- Java enqueue server
- Java message server

To enable transparent failover of the Java SCS to another system, the same considerations as for the ASCS apply (see “ABAP SAP Central Services (ASCS)” on page 136).

The individual Java SCS resources are started and stopped by the same mechanisms, using the same scripts as for the ASCS resources (see “ABAP SAP Central Services (ASCS)” on page 136).

ABAP application server instances

Read the provided information about shell scripts that allow you to start, stop, and monitor remote ABAP application server instances.

For an implementation of high availability Option 1 (see “Option 1: Automation using System Automation for z/OS only” on page 24), you need the following shell scripts that allow you to start, stop and check remote ABAP application server instances.

More granular monitoring and automation capabilities can only be achieved with HA Option 2 (see “Option 2: Automation by System Automation products with central components on AIX or Linux” on page 26), where all central components are running on AIX or Linux) which implies the deployment of Tivoli System Automation for Multiplatforms.

start_as <Hostname> <InstDir> <InstType> <jobname> <maxretries> [<via>]
Starts an ABAP application server instance, if <InstType> value of 0 is specified.

<jobname> is a unique job name for the **start_as** shell script. You must define a job name which is different to the job name listed under

Application Information of the SA z/OS proxy resource. This is required to avoid a false SA z/OS reaction on the pre-process termination exit when the script ends.

stop_as <Hostname> <InstDir> [<via>]

Stops an ABAP application server instance.

check_as <Hostname> <InstDir> <InstType>

Starts an ABAP application server monitor, if <InstType> value of 0 is specified.

These shell scripts and their parameters are described in detail in Chapter 14, “Reference of the z/OS high availability scripts,” on page 285.

If you are running an SAP NetWeaver kernel greater or equal to 7.38, then implement the changes described in “Required changes to protected web methods of sapstartsrv” on page 286.

Note: For installations with virtual host names, the parameter <Hostname> must be set to the virtual host name of the application server system.

The parameter <via> is optional. It identifies the remote execution type used to send commands to remote application servers (running under AIX, Linux on z Systems, or Windows). If a remote application server is started or stopped, the default is SSH. So it can be set to SSH or can be omitted.

What the shell scripts do

This information unit describes the tasks performed by the shell scripts.

start_as

- For the remote application server, the start_as script first checks if the remote host can be reached via ping. This is retried three times, while sleeping 10 seconds between attempts, so the total *ping wait* time is 30 seconds. After start_as can ping the host, it determines the type of the application server, which can be either AIX, Linux, or Windows. Then it checks for an ABAP application server, whether the database server can be reached from there via R3trans. In case of an error, the shell script indicates the status by sending a message to the system console, and then ends. Otherwise, if no error occurs, it then checks whether the instance is already running by using the IBM utility samsapctrl_asking (see section **samsapctrl_asking** in this information unit). If the instance is running, the shell script indicates the status by sending a message to the system console, and then ends.

This step protects a running application server instance from unnecessary restarts. For example, in case of an intermittent communication error, check_as terminates and Tivoli System Automation issues the start_as command again. Based on the notification of the active state message, Tivoli System Automation now starts check_as again.

Using this approach, Tivoli System Automation only has to monitor a single process, namely the one started by check_as.

- An AIX or Linux ABAP or dual-stack application server is started by invoking the following SAP standard scripts or commands:

```
stopsap r3 <instance> <hostname>
cleanipc <instnr> remove
startsap r3 <instance> <hostname>
```

The stopsap and cleanipc 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.

- Finally, the script checks periodically until the application server instance is running and responding to samsapctrl_asping. You can configure the maximum number of periodical checks using parameter <maxretries>. If the first check is unsuccessful the script retries for 10 * <maxretries> seconds. Successful startup is indicated by sending a message to the system console.

Example: With <maxretries> = 24, the script gives up after 240 seconds and returns a STARTUP FAILED message. If starting the Java application server takes longer in your environment, you must use a higher<maxretries> counter.

- An AIX or Linux Java application server is started by invoking the following SAP standard scripts or commands:

```
stopsap j2ee <instance> <hostname>
cleanipc <instnr> remove
startsap j2ee <instance> <hostname>
```

- A Windows application server is started with following SAP standard executables:

```
stopsap name=<SID> nr=<instance number> SAPDIAHOST=<hostname>
startsap name=<SID> nr=<instance number> SAPDIAHOST=<hostname>
```

stop_as

- The ABAP application server is stopped by invoking the following script for AIX and Linux:

```
stopsap r3 <instance> <hostname>
```

For Windows, the ABAP application server is stopped by invoking the following SAP executable:

```
stopsap name=<SID> nr=<instance number> SAPDIAHOST=<hostname>
```

check_as

- The health check of the application server is done by requesting health status information via the sapcontrol interface of the AS instance. The sapcontrol interface is periodically called using the samsapctrl_asping utility (see section **samsapctrl_asping** in this information unit). The parameter <InstType> specifies, for which application server type the monitor is started. A value 0 starts an ABAP only application server(AS) monitor, a value of 1 starts a dual-stack AS monitor, and a value of 2 starts a Java only AS monitor.

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

The script creates a link to the samsapctrl_asping executable such as ./AS_ABAP_ping_<hostname>_<instance number> in the current directory for an ABAP application server, or ./AS_Java_ping_<hostname>_<instance number> in the current directory for a Java application server.

In the SA z/OS policy for the respective resource application, edit the z/OS UNIX Control options. For correct monitoring via SA z/OS:

- the z/OS UNIX path should be /bin/sh
- the z/OS UNIX filter should be ./AS_ABAP_ping_<hostname>_<instance number> for an ABAP application server, or ./AS_Java_ping_<hostname>_<instance number> for a Java application server.

samsapctrl_asping

- This utility uses the SAP's sapcontrol interface of the application server instance to get its health status. The command line parameters allow you to choose between different modes. It must be installed in the home directory of the <sid>adm user.
- The default option is that samsapctrl_asping ends after it gets a response from the sapcontrol call for the ABAP or Java application server instance. This is used in the start_as script to check whether an application server instance is up and running.
- Another option specifies that samsapctrl_asping continues to run and checks the status periodically (the default is 30 seconds) via sapcontrol calls. It only ends if sapcontrol indicates a bad health status of the application server. This mode is used in the check_as script to monitor an ABAP or Java application server instance.
- samsapctrl_asping uses a message file. This file has the name *sapolicy.sap.map_USS*. It must be installed in the same directory as the samsapctrl_asping script, namely in the home directory of the <sid>adm user.

Invoking remote execution

SSH is the **only** remote execution type for the *start* and *stop* scripts that are used to control remote application servers. OpenSSH is the implementation of a secure shell which allows different methods of authentication. It is available as a Program Product for z/OS and as an Open Source product on most other platforms including Linux on z Systems and AIX. Under Windows, the scripts have been tested using the *Tectia SSH Client/Server* product, version 6.4, from the *SSH Communications Security* company as a sample SSH client/server implementation.

In the start_as script, the remote execution command is run in background.

z/OS V2.R2 changes to OpenSSH / IBM Ported Tools: Before z/OS V2R2 and IBM Ported Tools for z/OS 1.3, the current OpenSSH version was 5.0p1. It was available from IBM Ported Tools for z/OS .

Starting with z/OS V2R2 and IBM Ported Tools for z/OS 1.3, OpenSSH is version 6.4p1 and is now available as a base element of z/OS. Consider the z/OS migration information if you were using IBM Ported Tools before z/OS 2.2.

For more detailed information, refer to the following web sites and documentation:

- <http://www.ibm.com/systems/z/os/zos/features/unix/ported/>
- <http://www.openssh.org/>
- z/OS OpenSSH User's Guide
- Migration from z/OS V2R1 and z/OS V1R13 to z/OS V2R2

Instructions concerning the ssh-rand-helper command

- **Starting with z/OS 2.2**, the software **ssh-rand-helper** is longer supported. ICSF is required and must be started with/dev/random support prior to starting OpenSSH.
- **Prior to z/OS 2.2**, it is highly recommended that you avoid the use of the **ssh-rand-helper** command by using a hardware solution.

There is a known and intermittent issue with ssh-rand-helper, which even may prevent you from stopping the SAP application servers. This behavior may be accompanied by the following messages:

FOTS1842 problem renaming PRNG seedfile from /u/haladm/.ssh/prng_seed.i184017545 to /u/haladm/.ssh/prng_seed (EDC5129I No such file or directory.)
FOTS1945 ssh-rand-helper child produced insufficient data

If possible, avoid using ssh-rand-helper. If you have a cryptographic coprocessor, you can enable /dev/random in z/OS UNIX. This improves start-up times and reliability. See topic *Using hardware support to generate random numbers* in the IBM Ported Tools for z/OS: OpenSSH User's Guide for more information.

Java and dual-stack application server instances

This topic provides information about shell scripts that allow you to start, stop, and check remote Java application server instances.

To start and check remote Java application server instances from z/OS UNIX, the previously mentioned shell scripts are used (see “What the shell scripts do” on page 138):

- start_as

If start_as is invoked with <InstType> set to 1, it starts a dual-stack application server (AS) and a value of 2 starts a Java only application server.

- check_as

If check_as is invoked with <InstType> set to 1, it starts a dual-stack application server (AS) and a value of 2 starts a Java only application server.

In the **SAPSRV add-on policy*, the SAP application server instances are modeled as **proxy** resources (see “SAP application servers as proxy resources” on page 171). The start/stop of the real resources is done via SSH. Monitoring is done with the samsapctrl_asping utility.

Proxy resource also implies that a normal stop of the application server resource STOP NORM under SA z/OS only stops the monitoring utility for that resource. If you want to stop the resource itself, you must use the STOP FORCE of SA z/OS. This also stops the application server on the remote host.

With SA z/OS 3.5 APAR level OA48922, the stop policy of proxy or remote application server resources has changed. For SA z/OS operators, this new policy eliminates the need to explicitly use the STOP FORCE mode or the SHUT FORCE phase to really stop the application server itself.

Note that you can also use this option for proxy or remote application server resources created with an older version of the **SAPSRV add-on policy*. In that case you must manually change the stop policy as described with the application server specific parameters in Chapter 8, “Customizing Tivoli System Automation for z/OS,” on page 145.

The new stop policy uses the new REXX script SAPRASTP. This script automatically decides when to use STOP NORMAL or STOP FORCE mode. The decision is based on the *STOP votes* existing in SA z/OS for the proxy resource:

- If a *STOP vote* with desired state UNAVAILABLE for the *Move group* exists, then the remote application server is stopped.
- Otherwise only the proxy resource is stopped and automatically restarted on a different LPAR. This suggests that the stop was for an LPAR maintenance scenario.

It is also important to understand that a dual-stack application server is physically *one SAP instance*. This instance runs both the ABAP and the Java stack. Although it

is physically one instance and both stacks are started when the (ABAP) instance starts, the SA policy separates it into its *two* logical parts:

- the ABAP application server
- the Java application server

This means that within System Automation, one dual-stack application server instance is automated as *two* logical application server instances:

- an ABAP application server instance
- a Java application server instance

However, there is a close relationship between these two logical application servers:

- The Java instance must only be started *after* the ABAP instance is active. So there is a *HasParent* relationship between them. This *HasParent* relationship guarantees that starting the Java instance *automatically* triggers the start of the ABAP instance before. As a result, the start will simply wait until Java is up.
- Stopping the Java application server does *not* stop any of the Java server processes. It only stops the monitoring `samsapctrl_asping` utility, which does a health check of the Java application server.

For a description of the implementation of this relationship in the **SAPSRV add-on policy*, refer to “SAP application servers as proxy resources” on page 171.

Note: The related `stop_as` script is described in Chapter 14, “Reference of the z/OS high availability scripts,” on page 285. It is used in the same way as for stopping ABAP instances.

SAP host agent

SAP requires an SAP host agent to be installed and running on each z/OS LPAR on which SAP components can run. This does not only include LPARs on which SAP Central Services are installed, but also LPARs on which just a DB2 subsystem for SAP is installed. This information describes installation and automation considerations for an SAP host agent.

Whenever an SAP component - for example an SAP central services instance - is installed, the SAP installation tool either:

- installs an SAP host agent, if no installation is found on the LPAR
- updates an existing SAP host agent, if a newer version is available.

The standalone installation procedure is described in the SAP host agent documentation. See *SAP Note 1907566: Obtaining the Latest SAP Host Agent Documentation* and *SAP Note 1031096: Installing Package SAPHOSTAGENT*.

The installation procedure places the SAP host agent into the SAP system independent path `/usr/sap/hostctrl`. This directory must be created in an zFS file system, which is local to the host LPAR, as described in “SAP host agent file system” on page 100.

When the SAP host agent is started on an LPAR under z/OS UNIX, you see its two main processes `sapstartsrv` and `saphostexec`. These processes are modeled as resources `SAPHOST_CTL` and `SAPHOST_EXE` in the **SAPSRV add-on policy*.

With the default setup, the SAP host agent automatically starts a `saposcol` process when it starts on an LPAR. The `saposcol` process is controlled (started and stopped) by the SAP host agent. Therefore, `saposcol` is not modeled as a separate resource in the **SAPSRV add-on policy*.

In new versions of the SAP host agent, the CIM-based monitoring agent `sapcimb` is delivered. It can be used as an alternative to `saposcol`. Currently, both `saposcol` and `sapcimb` are supported by SAP. For a description of `sapcimb` on z/OS UNIX System Services, see *SAP Note 2047924: DB2-z/OS:CCMS:HAG: CIM-Provider Enablement for z/OS*.

Similar to `saposcol`, the `sapcimb` process is started and stopped by the SAP host agent, therefore `sapcimb` is not modeled as a separate resource in the **SAPSRV add-on policy* either.

If you define a dynamic VIPA for the RMF DDS (see “Configuring Sysplex performance monitoring highly available with the RMF Distributed Data Server (RMF DDS)” on page 104), then you must ensure that the environment variables `SAP_IBMZMON_DDS_HOST` and `SAP_IBMZMON_DDS_PORT` are set in the environment of the `saproot` user to the dynamic VIPA host name and the DDS port when starting the SAP host agent. Since the SA z/OS policy uses the CSH shell to start the SAP host agent, you must ensure that these environment settings are also made in the `cs` environment, for example by setting these variables in the `.cshrc` file of the `saproot` user.

SAProuter

The SAProuter process is started and stopped by Tivoli System Automation directly by starting the SAP executable - no additional shell script is needed.

The SAProuter start command does not specify the name of the route permission file with the `-R` option. Therefore, if you do not want to use the default route permission file, you need to adapt the start command of the SAProuter resource `SAPSYSRT` in the **SAPSRV add-on policy* and add the `-R routtab` option. Otherwise, the default route permission file `./saprountab` in the home directory of the user who starts the SAProuter, is used. For example, if started from user `ha2adm`, then the route permission file `/u/ha2adm/saprountab` is used.

Summary of start, stop, and monitoring commands

A table in this section summarizes the start, stop, and monitoring commands that are needed when you set up the Tivoli System Automation policies for SAP.

Table 13. Summary of start/stop monitoring commands

Actions	Value or command
VIPA for ASCS and SAProuter: - start command (started task)	S INGEVIPA,VIPA=172.20.10.1

Table 13. Summary of start/stop monitoring commands (continued)

Actions	Value or command
ASCS, SCS, ERS, WD, or SMDA	
- start command	/u/<sid>adm/start_cs <WD SID> <WD Instance Name> <Name of WD VIPA> <WD JobName> <maxRetries>
- stop command	sapcontrol -nr <WD Instance Number> -function Stop
- process name to be monitored	./wdispmon_<LPAR>_<SID>_<WD Instance Name>_<Name of WD VIPA>
- monitor command	/u/<sid>adm/checkwd <LPAR Name> <WD SID> <WD Instance Name> <Name of WD VIPA> Note: <WD SID>, <WD Instance Number>, and <WD Instance Name> stand as examples for the SID and instance number and name of ASCS, SCS, ERS, WD, or SMDA
Application server instances:	
- start command	/u/<sid>adm/start_as <Hostname> <InstDir> <InstType> <jobname> <maxretries> [(via)]
- poststart (monitor) command	/u/<sid>adm/check_as <Hostname> <InstDir> <InstType>
- stop command	/u/<sid>adm/stop_as <Hostname> <InstDir> [(via)]
- process name to be monitored	/u/<sid>adm/AS_ABAP_ping_<Hostname><InstNr> (ABAP app.server) /u/<sid>adm/AS_Java_ping_<Hostname><InstNr> (Java app.server)

Note:

1. ABAP and Java SCS are started by the start_cs script script. See “Sample scripts” on page 290 for details and the invocation syntax.
2. Since version 3.3, System Automation for z/OS, SA ships the sample procedure INGEVIPA which is used in Table 13 on page 143 to start the VIPAs .

Chapter 8. Customizing Tivoli System Automation for z/OS

Read the contained information on how to set up IBM Tivoli System Automation for z/OS (SA z/OS) for the high availability solution for SAP, based on the **SAPSRV add-on policy*.

Availability and use of the current **SAPSRV add-on policy*

This edition of *Business Continuity for SAP on IBM z Systems* is based on the **SAPSRV add-on policy* of SA z/OS 3.5, APAR level OA51440. The previous **SAP add-on policy* in earlier System Automation releases is no longer part of the System Automation product.

If you have migrated your SA policy from earlier releases, and if this policy contains SAP systems that are automated with the previous **SAP add-on policy*, then you can continue to use it on SA 3.5.

The **SAPSRV add-on policy* uses the standard SAP `sapstartsrv` service for starting SAP instances. Thereby it enables seamless integration of an SA z/OS controlled SAP Central Services instance into SAP systems management tools, like SAP's Management Console, or into SAP life cycle tools, like the SAP Software Update Manager (SUM). Furthermore, the use of this policy is a prerequisite for using the SAP HA Interface (see "SAP HA Interface for SA z/OS" on page 168).

Note:

1. Although the SAP HA wizard can significantly reduce the effort for creating a SA policy for SAP, a detailed knowledge of IBM Tivoli System Automation for z/OS is nevertheless required to make SAP high availability work.
2. Throughout this publication, the sample SAP system ID (SID/SAPSID)
 - of *SID* is used for all resources that belong to SAP Central Services instances
 - of *CON* is used for the resources of an optional SAP Web Dispatcher
 - of *SM1*, *SM2*, and *SM3* are used for the resources of optional SAP Solution Manager Diagnostics Agent (SMDA) installations.

The SAP HA wizard replaces these sample *<SID>*s with the actual SID of your SAP installations. If you create your SA policy without using the wizard, you need to manually replace the sample values with the SIDs (SAPSIDs) of your own SAP installation.

Migration and coexistence

Policy definitions for new SAP systems should preferably be made based on the current **SAPSRV add-on policy*. After generating such a policy, you can import these resources into an existing policy, which contains resources for other SAP systems that were modeled based on the old **SAP add-on policy*. This import is possible because the only overlap is in the common class `C_SAP_USS` - which did not change from the **SAP* to the **SAPSRV add-on policy*.

For SAP systems that are automated using the old **SAP add-on policy*, you should consider re-generating the policy based on the current **SAPSRV add-on policy* to benefit from the better integration with previously mentioned SAP tools. The SAP

HA wizard (see “How to adapt the SA z/OS **SAPSRV add-on policy*” on page 149) enables the quick and easy generation of such a policy.

To facilitate the migration from SA z/OS 3.4 to 3.5 you might consider running a mixed SA 3.5 and SA 3.4 environment - which is supported by the System Automation product. For details see topic *Planning --> Migration Information --> Coexistence of SA z/OS 3.5 with Previous Releases* from the IBM Knowledge Center for Tivoli System Automation for z/OS 3.5.0.

Features of the **SAPSRV add-on policy*

The **SAPSRV add-on policy* assumes that if you use an enqueue replication server, you installed it with its own virtual host name. For a rationale, see “Virtual host name for the ERS instance” on page 118.

Additionally, the **SAPSRV add-on policy* also models the SAP host agent.

This **SAPSRV add-on policy* is supported by the *SAP HA wizard*, which is included in the tar-file `ING_sap.tar` in the z/OS UNIX directory: `/usr/lpp/ing/SAP`.

The SAP HA wizard uses the **SAPSRV add-on policy*. You can interactively construct a new SAP policy. For a detailed description of the wizard operation, see the PDF file included in the tar-file.

The **SAPSRV add-on policy* includes resource definitions for remote application server resources. The SAP HA wizard supports the adaption of these resources to your specific SID. See “SAP application servers as proxy resources” on page 171 for a description of these resources.

Note: The **SAPSRV add-on policy* comes with new job names for all resources compared to the **SAP add-on policy*. See Table 9 on page 93 for details. You may need to adapt port bindings within the `TCPIP.PROFILE`, if you use this feature.

Prerequisites

The following assumes that you set up your environment, as described in Chapter 5, “Concepts for a high availability SAP solution,” on page 69 through Chapter 7, “Customizing SAP for high availability,” on page 113. The **SAPSRV add-on policy* is based on SAP Kernel 7.20 or higher. As a consequence, SAP Syslog Collector and Sender are no longer necessary and supported. For details on how to use the SAP central system log via HTTP or HTTPS, refer to *SAP Notes*[®] [1704753](#), [1041390](#) and [1636252](#) in Table 47 on page 320.

Preparing SA z/OS for SAP high availability

This topic discusses what you need to do before you define the SAP-related components in the SA z/OS policy.

Before you start to customize your SA z/OS policy for the high availability solution, make sure that the basic installation and customization of NetView and SA z/OS is complete.

The base z/OS resources must be defined to SA z/OS. Here is an excerpt of the very basic ones:

- JES

- NetView, NetView Subsystem Interface and NetView UNIX Server
- OMPROUTE (**Note:** If an AUTOLOG statement is defined in the TCPIP profile for OMPROUTE, you must change this to NOAUTOLOG)
- RPCBIND (**Note:** If an AUTOLOG statement is defined in the TCPIP profile for RPCBIND, you must change this to NOAUTOLOG)
- RRS
- TCP/IP and VTAM®
- For further z/OS resources refer to the *BASE add-on policy that is shipped with the SA z/OS product.

If you use any other automation tool, for example, Automated Restart Manager (ARM) in your environment, your configuration needs to be checked to ensure that it does not interfere with Tivoli System Automation. See also “Policy changes for TCPIP and OMPROUTE” on page 148.

If you want to make use of the Status Display Facility (SDF) or Tivoli Enterprise Portal (TEP), make sure it is customized and working.

You might consider installing **Service Management Unite**, which provides a web GUI for System Automation. For details see *IBM Service Management Unite*.

In **Service Management Unite**, you can create and customize graphical dashboards that are specialized for your SAP environment. For examples, see this article on developerWorks:

Service Management Unite Dashboards for SAP

Make sure that SA z/OS starts and stops all applications and puts them into a satisfactory status of availability, which shows the application resource in a green color.

How to send UNIX messages to the z/OS syslog

Follow the steps that are required to enable the sending of UNIX messages to the z/OS syslog.

To enable the sending of UNIX messages to the z/OS syslog, proceed as described in the listed steps.

1. Ensure that you have a running z/OS UNIX System Services syslog daemon **syslogd**. Information on the control and configuration is contained in the following manuals:
 - *z/OS UNIX System Services Planning*
 - *z/OS Communication Server IP Guide*
2. To enable UNIX **syslogd** to send messages to the z/OS syslog, add the following entry to your syslog configuration file `/etc/syslog.conf` (or to the file specified on the start of the **syslogd** daemon with the `-f` option):

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

UNIX messages are displayed in the z/OS syslog with a BPXF024I message-ID. These messages (that are sent to the console) are important for the operation of SA policies if you want to automate remote application server resources.

RPCBIND and NFS server - SA z/OS relationships in network configurations with and without OMPROUTE

Modify the relationships between the RPCBIND and NFS server in your network configuration to ensure a continuous availability of the NFS server.

The NFS server uses the RPC protocol. The IBM strategic direction for RPC support is the RPCBIND server. It has no upper limit for handling parallel requests, which makes it ideally suited to support the NFS server. For a presentation about the features of RPCBIND, see the IBM education assistant for z/OS Communications Server Version 1 Release 8.

The SA z/OS **USS add-on policy* and the **BASE add-on policy* include definitions of resources and relationships for NFSSERV, RPCBIND, TCPIP, OMPROUTE. Relationships can define a dependency together with a condition (*ForceDown/WhenObservedHardDown*) or without a condition (*HasParent*). Verify that the required relationships exist in your policy. Otherwise, modify the policy according to the following instructions:

With OMPROUTE

Add the following relationships, if they are missing, so that the NFS server can move to another LPAR immediately when OMPROUTE goes into HARDDOWN:

- At the NFSSERV resource, add a *HasParent* relationship to both RPCBIND and OMPROUTE.
- At the NFSSERV resource, also add a *ForceDown/WhenObservedHardDown* relationship to both RPCBIND and OMPROUTE.

Without OMPROUTE

Add the following relationships, if they are missing, so that the NFS server can stay on the LPAR when TCPIP and RPCBIND recover from a failure:

- At the NFSSERV resource, add a *HasParent* relationship to both RPCBIND and TCPIP.
- At the NFSSERV resource, also add a *ForceDown/WhenObservedHardDown* relationship to RPCBIND.

If you need information on how to move from PORTMAP to RPCBIND, see “From PORTMAP to RPCBIND” on page 311.

Policy changes for TCPIP and OMPROUTE

If you use OSPF for dynamic routing implementation with OMPROUTE, read here what you need to observe in such a case.

If you automate SAP on z Systems with System Automation for z/OS and use OSPF for dynamic routing implementation with OMPROUTE, then you must set the message EZD1214I as the ONLY TCPIP UP message (EZD1214I INITIAL DYNAMIC VIPA PROCESSING HAS COMPLETED FOR TCPIP).

If you additionally use:

GLOBALCONFIG SYSPLEXMONITOR *MONINTERFACE DYNROUTE*

you must use following additional options:

- DELAYJOIN
- NORECOVERY

DELAYJOIN

This option causes the sysplex monitor to keep TCPIP from joining the sysplex until OMPROUTE has dynamic routes and you have OMPROUTE for that reason. But this option also prevents the stack from joining the sysplex (and by extension, issuing the EZD1214I message) until OMPROUTE registers with the stack. So if your automation is set to trigger the start of OMPROUTE on TCPIP being UP, you created a deadlock.

Therefore, OMPROUTE must be defined in the SA z/OS policy with option *external startup* set to initial, and TCPIP must start OMPROUTE via the AUTOLOG statement. Additionally all SAP resources must have an extra *HasParent* relationship to TCPIP.

NORECOVERY (default value)

When a problem with TCPIP is detected, issue messages that are related to the problem but take no further action. This is what you need, as TCPIP's start, stop, and restart is controlled by System Automation for z/OS and SYSPLEXMONITOR must not interfere with it.

Policy changes for EnqCF replication

If you use the enqueue replication into an IBM z Systems coupling facility, you must apply required changes in the policy to allow a restart of the enqueue server on the same LPAR after a critical failure.

To allow a restart of the enqueue server in place on the same LPAR after a CRITICAL failure, you need to change the CRITICAL THRESHOLD for the ENQ resource in the SA z/OS policy to a value greater than 1. The recommended value is 5, or a different value which is less than the restart attempts from sapstartsrv (which defaults to 6). If you use the SAP HA wizard of SA z/OS APAR level OA47266 or higher to generate an SCS policy, then the wizard generates a policy that already contains this change.

How to adapt the SA z/OS **SAPSRV add-on policy*

Read the described instructions and hints on what to observe when you need to adapt the **SAPSRV add-on policy* to your environment.

To adapt the **SAPSRV add-on policy* that is shipped with SA z/OS, it is recommended that you use the SAP High Availability wizard (SAP HA wizard). This wizard can generate an SA z/OS policy for a new SAP system. It also automates the bulk of the renaming and adaptation. For details about how to use the SAP HA wizard, see *SAPHAwizard.pdf*, which is included inside the ING_sap.tar file in the SA z/OS UNIX System Services directory /usr/lpp/ing/SAP.

ABAP and Java SCS are started by the start_cs script. It uses the sapcontrol executable, which is delivered by SAP to start the resources of the ABAP and Java SCS instances: enqueue server (ES), message server (MS), enqueue replication server (ERS), and gateway server (GW). With the **SAPSRV add-on policy*, the central services components (EN, MS, GW, ERS) are no longer started directly as with the **SAP add-on policy*. The sapcontrol executable issues a start request to the sapstartsrv service. The sapstartsrv service starts a sapstart process, which in turn finally starts the previously mentioned processes.

The sapstartsrv service that is associated with the ABAP and Java SCS is started by the start_sapsrv script. It also uses the sapcontrol executable that is delivered

by SAP to start the `sapstartsrv` service of the associated ABAP and Java SCS instance. Both scripts are contained in the HA wizard tar-file and you must copy them to the home directory of the `<sid>adm` user.

If you implemented a network setup that is not based on dynamic routing as described in “Alternative network setup” on page 54, you must replace the dependencies to the `OMPROUTE` resource:

- Replace the *HasParent* relationship of the SAP and NFS server resources to the `OMPROUTE` application by a *HasParent* to `TCPIP`
- Change the *ForceDown* relationship of the SAP and NFS server resources to the `OMPROUTE` application to a *ForceDown/WhenObservedAssumedDownOrStopping* dependency to the `TCPIP` application

Overview of the add-on policy for SAP

This information together with the contained subtopics provide an understanding of the resources that are defined in the **SAPSRV add-on policy*. With this knowledge, you can determine, which of these resources are required for your specific environment.

The following SAP-related components are defined in the SA z/OS **SAPSRV add-on policy*. Not all of the components might be needed for your specific environment. For example, for an ABAP-only SAP system you do not need the Java components. Therefore the first step in adapting the policy for a specific SAP system is to determine which parts of the policy you actually want to exploit.

The SAP HA wizard detects, based on the available SAP configuration files, which SAP components are to be automated and it generates a policy that contains these components.

The SA resources that might be relevant for an SAP system can be classified as follows:

- Resources that are related to a specific SAP system:
 - DB2 z/OS resources
 - ABAP central services - including enqueue server and message server, and the associated VIPA.
 - ABAP enqueue replication server and the associated ABAP ERS VIPA
 - Java central services - including enqueue server and message server, and the associated VIPA
 - Java enqueue replication server and the associated JAVA ERS VIPA.
 - SAP Java Gateway
 - With SAP 7.1 and higher, the SAP Java SCS has its own standalone gateway process that is modeled as a resource in the SA z/OS **SAPSRV add-on policy*
 - Remote SAP application servers
- Resources that are common to all the SAP systems:
 - NFS server
 - SAProuter or SAP Web Dispatcher or both
 - SAP Host Agent executables to monitor a host (LPAR) with SAP NetWeaver components
 - SAP Solution Manager Diagnostics Agent (SMDA).

Resource naming conventions

Certain conventions are used in applying entry names and subsystem or automation names to the SAP resources.

For all types of resources, SA z/OS uses entry names in its dialog interface on the one hand. In the automation processing, however, it uses subsystem names for classes and applications, and automation names for groups. Users must specify all names when they define the resources. However, entry names can be longer than subsystem names or automation names. For SAP resources referenced in this publication, normally both names are equal. The presented graphics show the resource entry names only. In the text, the automation name or subsystem name of a resource is shown in brackets, if they are different from the entry name.

To retrieve the real SAP resource names, you must replace the placeholder <SID> from the names shown in Table 14 with the current SAP system ID of a concrete SAP system. For the same purpose, you must also replace the placeholder <CON> with the SAP system ID of a concrete SAP Web Dispatcher system.

Table 14. Differences between entry names and subsystem or automation names

Entry name	Automation/Subsystem name
<i>Classes with entry names and different subsystem names</i>	
C_SAP_<SID>_BASE	C_SAP_<SID>_B
C_SAP_<SID>_ABAP	C_SAP_<SID>_A
C_SAP_<SID>_JAVA	C_SAP_<SID>_J
C_SAP_<CON>_WEBD	C_SAP_<CON>_W
C_SAP_SM<n>_SMDA (n = 1,2,3)	C_SAP_SM<n>_S (n = 1,2,3)
<i>Applications with entry names and different subsystem names</i>	
SAPHOST_EXEC	SAPHOST_EXE
SAP<CON>WD_SRV	SAP<CON>WD_SR
SAP<CON>WD_STA	SAP<CON>WD_ST
SAP<SID>A_STA	SAP<SID>AST
SAP<SID>AR_SRV	SAP<SID>AR_SR
SAP<SID>AR_STA	SAP<SID>ARST
SAP<SID>J_STA	SAP<SID>JST
SAP<SID>JR_SRV	SAP<SID>JR_SR
SAP<SID>JR_STA	SAP<SID>JRST
SAPSM<n>SM_SRV (n = 1,2,3)	SAPSM<n>SM_SR (n = 1,2,3)
SAPSM<n>SM_STA (n = 1,2,3)	SAPSM<n>SM_ST (n = 1,2,3)
<i>Groups with entry names and different automation names</i>	
SAPHOST_AGENT	SAPHOST_AGT
SAP<CON>WDSRVX	S<CON>WDSRV_X
SAP<SID>ASRV_X	S<SID>A_SRV_X
SAP<SID>ARSRVX	S<SID>ARSRV_X
SAP<SID>JSRV_X	S<SID>J_SRV_X
SAP<SID>JRSRVX	S<SID>JRSRV_X

Group structure

Groups are defined in a hierarchy in the **SAPSRV add-on policy*.

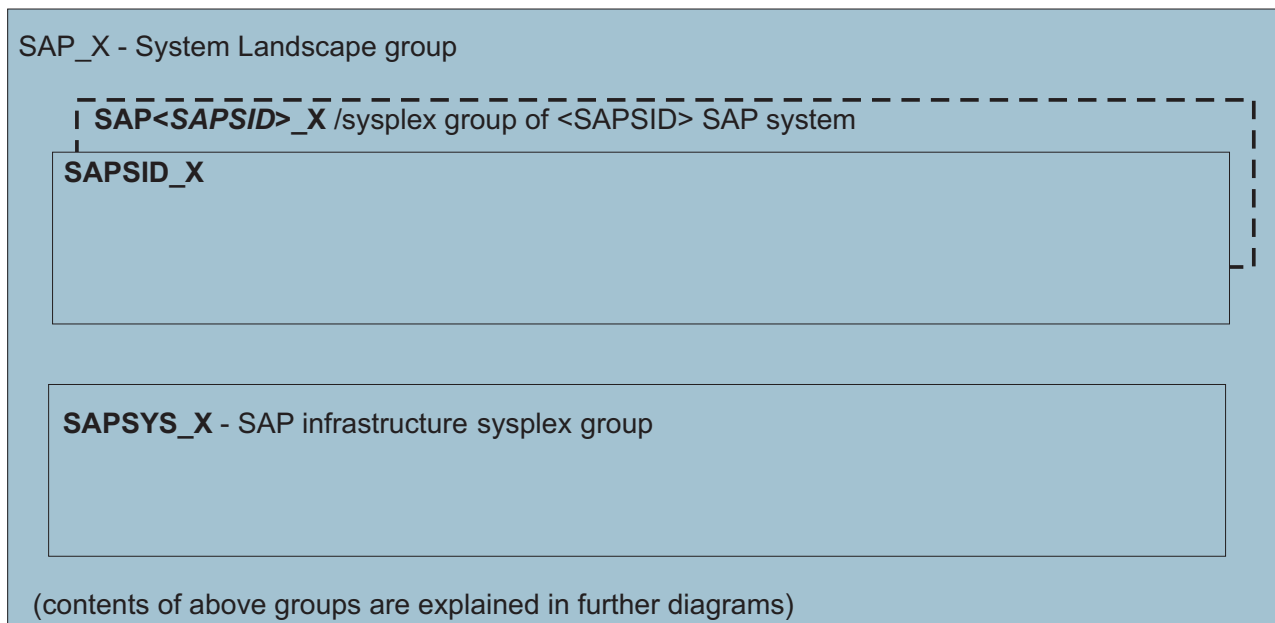
The top-level group in the **SAPSRV add-on policy*, which contains all the components that are listed in “Overview of the add-on policy for SAP” on page 150, is SAP_X. Figure 26 shows the SAP_X group, which contains

SAPSYS_X

The group that contains all SAP-system-independent groups and resources.

SAP<SID>_X

The group that contains specific groups and resources for SAP system <SID>.



----- indicates optional resources / optional groups

■ Sysplex/Basic group

Figure 26. SA z/OS **SAPSRV add-on policy* for SAP

In your SA z/OS production policy database, the SAP_X group contains all SAP<SID>_X groups for all SAP systems with resources that are managed by System Automation.

Note: You can use the SAP_X group for easy monitoring of the availability of your *complete SAP landscape*. If this group is available, your SAP systems are also available.

Class structure

This topic explains the purpose of classes and contains information about the class hierarchy for the SAP resources that are defined in the **SAPSRV add-on policy*.

Figure 27 on page 154 shows the class hierarchy for the SAP resources.

A *class* in SA z/OS can be used to define common characteristics, which are shared by many resources. The **SAPSRV add-on policy* makes intensive use of the class concept in SA z/OS.

Like in the old **SAP add-on policy*, C_SAP_USS is the top-level class for all SAP resources. Since this class was not changed between the **SAP add-on policy* and the **SAPSRV add-on policy*, you do not need to replace it, if your active production PDB already contains it. All SAP z/OS UNIX resources should refer to this class. It contains definitions, which are common to all SAP resources. It defines, for example, the RESTART OPTION as ALWAYS. This is because any abnormal end of a UNIX application appears to SA z/OS as a shutdown outside of the automation condition. The restart option ALWAYS enables SA z/OS to correctly recover from these situations.

Depending on your environment you might need to change definitions in this class. For example, on a slow system you might have to adjust the Start Delay and Start Cycles of your SAP z/OS UNIX resources.

The SA z/OS policy command line is restricted to 160 characters. Therefore, for example all ASCS or SCS resources, contain the tilde symbol (~) in the template of the command lines for starting and stopping the resources. The tilde saves a lot of space in that command lines compared to defining the home directory of the <sid>adm.

Below this top-level class the **SAPSRV add-on policy* contains classes, which are usage-specific (C_SAP_<SID>_ABAP or C_SAP_<SID>_JAVA, C_SAP_<SID>_BASE and C_SAP_<SID>_WEBD) and which depend on the specific SID of the resources that reference it. Most of the SID-specific information is contained in the symbol definitions of these classes.

The resource names and the values for the symbols in the **SAPSRV add-on policy* contain as SAP system ID (SID/SAPSID) a value of <SID> for the standard ABAP and Java central services components and <CON> for the optional SAP Web Dispatcher components.

These template classes from the **SAPSRV add-on policy* need to be converted into SID-specific classes as shown in Table 15 for the sample SIDs HA1 and WD2 for the SAP system and an SAP Web Dispatcher respectively. The class names are shown with their SA z/OS entry names. Their corresponding subsystem names are the same, without the last three letters, for example, C_SAP_HA1_B, see also Table 14 on page 151.

Table 15. SID-specific classes

Class name in *SAPSRV	Sample class name generated by wizard	Comment
C_SAP_<SID>_BASE	C_SAP_HA1_BASE	SAP Class for SAPSID-specific data, needed by ABAP or Java VIPA resources
C_SAP_<SID>_ABAP	C_SAP_HA1_ABAP	SAP Class for ABAP SAPSID-specific data, needed by ABAP resources
C_SAP_<SID>_JAVA	C_SAP_HA1_JAVA	SAP Class for Java SAPSID-specific data, needed by Java resources
C_SAP_<CON>_WEBD	C_SAP_WD2_WEBD	SAP Class for Web Dispatcher, needed by SAP Web Dispatcher resources

Table 15. SID-specific classes (continued)

Class name in *SAPSRV	Sample class name generated by wizard	Comment
C_SAP_SM<n>_SMDA	C_SAP_DA<n>_SMDA	SAP Class for Diagnostics Agent <n> (n = 1,2,3), needed by SAP Solution Manager Diagnostics Agent resource

The following figures show the class structure before (Figure 27) and after (Figure 28 on page 155) it was adapted to the specific SID HA1:

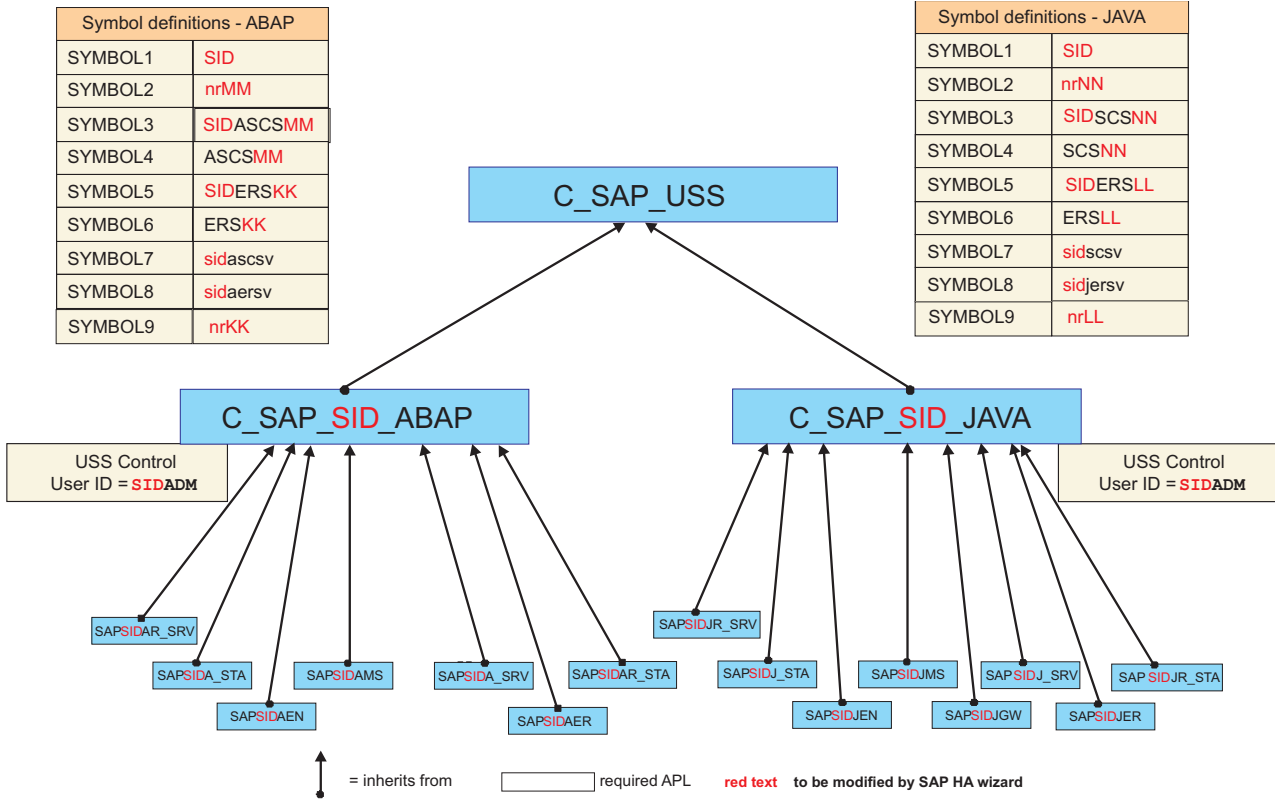


Figure 27. *SAPSRV add-on policy - class structure for SAP Central Services resources

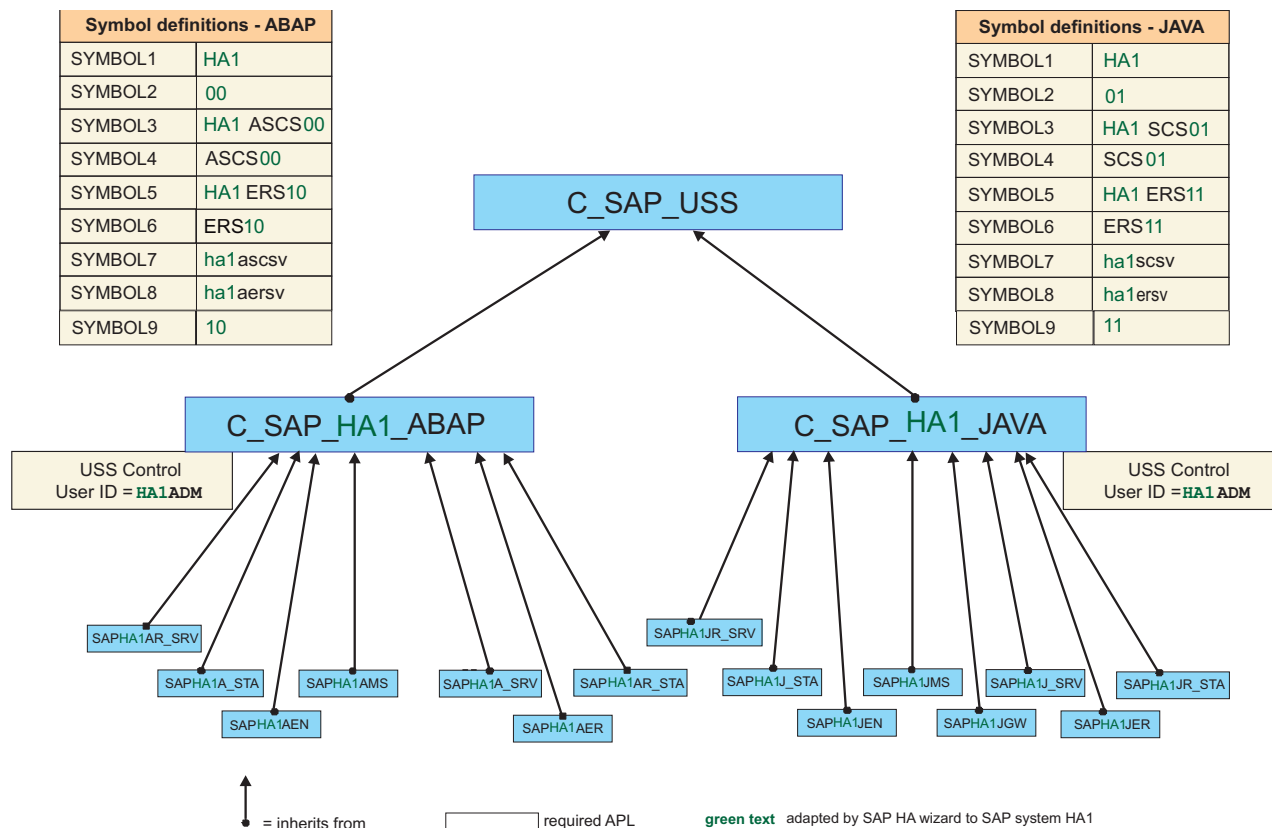


Figure 28. *SAPSRV add-on policy - class structure for HA1 sample system

For a complete list of symbols that are used in the *SAPSRV add-on policy, refer to the definition of this policy that is shipped with the SA z/OS product. Also refer to the file SAPHawizard.pdf, which is shipped inside the ING_sap.tar file in the SA z/OS z/OS UNIX directory /usr/lpp/ing/SAP.

SAP infrastructure group

View this information to learn about the purpose and contents of the SAP infrastructure group SAPSYS_X.

The SAP infrastructure group SAPSYS_X contains all groups which are needed for the operation of the SAP resources contained in all SAP<SID>_X groups. The recommendation for this group is that it includes the NFS server on z/OS, which ensures that this single point of failure for ALL SAP systems is highly available.

Figure 29 on page 156 shows the details of the SAP-system independent groups and resources. It contains these groups and resources:

- The NFS server group (NFS_SERV_X).
The NFS server resources are part of the SA z/OS *USS add-on policy and you might need to adapt it to your environment. Make sure that you have set up the NFS server/clients according to the specifications described in: Chapter 6, “Preparing a high availability SAP solution,” on page 89.
- The SAP Host Agent group (SAPHOST_AGENT [automation name SAPHOST_AGT]). This group contains the two SAP resources that model the SAP host agent. See “SAP host agent” on page 142.

Note: The SAP Host Agent exists only once on every z/OS LPAR. It does not depend on any specific SAP system, nevertheless it is installed (or, in most cases, updated to the newest level) with every SAP central services installation. When you run the SAP HA wizard for a specific SAP system, the SAP Host Agent resources are generated as well. You might not need these resources since they were already generated in a previous invocation, and are already part of your policy database. The SAP HA wizard offers to remove the Host Agent resource from the generated policy in its post-processing step.

- An optional group for the SAProuter (SAPSYSRTE_X).
- An optional group for the SAP-system-independent SAP Web Dispatcher group with an SAP system (SAPCONWDP_X).
- The SAP Solution Manager Diagnostics Agent (SMDA) resources.

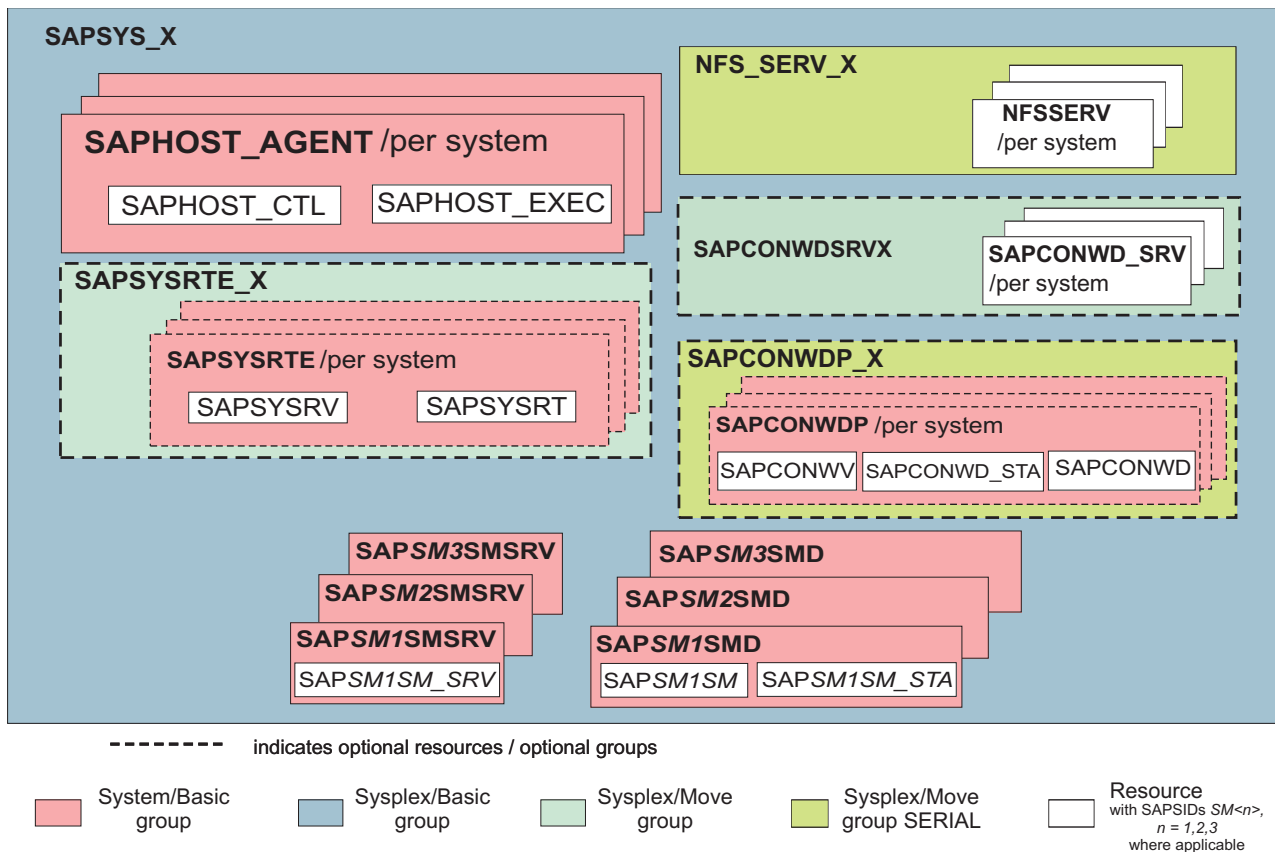


Figure 29. SAP-system-independent groups and resources

Further details of Figure 29:

• **The NFS server**

The NFS server that is needed for the SAP infrastructure must be active only on one LPAR in the SYSPLEX at a time. This is accomplished by defining a SYSPLEX MOVE group *NFS_SERV_X* with the attribute SERIAL. This group contains the NFS server *NFSSERV* as resource for each system. The *NFS_SERV_X* group is provided in the SA z/OS **USS add-on policy*.

Note: The SA z/OS **SAPSRV add-on policy* contains a reference to an *NFS_SERV_X* group in its *SAPSYS_X* group. If you are using a different name for your NFS Server group you must adapt this reference.

- **The SAP Host Agent group**

To enable monitoring of a host (LPAR) with SAP NetWeaver components, two executables must run on that host: SAPHOST_EXEC [subsystem name SAPHOST_EXE] and SAPHOST_CTL. The second is always started and stopped by the first one. SAPHOST_EXEC also starts a saposcol process. This saposcol process is no longer modelled in the **SAPSRV add-on policy*, because it is controlled by SAPHOST_EXEC. The SA policy writes two log files, when starting or stopping the resource:

- Stop log: /usr/sap/hostctrl/work/saphex_stop.C0H1.log
- Start log: /usr/sap/hostctrl/work/saphex.C0H1.log

- **The SAProuter group**

If you decide to run your SAProuter on z/OS, you also must define a VIPA to be used in accessing the SAProuter. The SAProuter and its associated VIPA must run together on the same LPAR. Therefore, the policy defines two SA applications SAPSYSRT and SAPSYSRV, which correspond to the SAProuter and its associated VIPA. Both are grouped in a SYSTEM group SAPSYSRTE. Since there must be only one active SAProuter and its associated VIPA in the SYSPLEX at a time, define the SYSPLEX/MOVE group SAPSYSRTE_X containing SAPSYSRTE (as shown in Figure 29 on page 156).

- **The SAP Web Dispatcher group**

- SAP Web Dispatcher installations that are based on SAP kernel 7.20 or higher may be used to serve multiple SAP systems (see “The SAP Web Dispatcher” on page 131). The **SAPSRV add-on policy* reflects this possibility by placing its SAPCONWDP_X and SAPCONWDSRVX SYSPLEX groups into the SAP infrastructure group SAPSYS_X.
- The SAP Web Dispatcher is installed under an SAP system ID of its own. In the SA z/OS **SAPSRV add-on policy*, it has the sample SID of CON, which needs to be adapted either by the SAP HA wizard or by a manual change to the installed SAP Web Dispatcher SID.
- If you decide to run your SAP Web Dispatcher on z/OS, you must define an own Web Dispatcher VIPA. As it is now modeled based on the sapstartsrv framework, three SA applications, SAPCONWD_STA, SAPCONWD, and SAPCONWV are defined. The first models the sapstart process, the second models the SAP Web Dispatcher process itself, and the third application models its associated VIPA. They are grouped in a SYSTEM group SAPCONWDP. The SAPCONWD_SRV application models the sapstartsrv process, which is in the SAPCONWDSRVX SYSPLEX group.

Important: If the specific Web Dispatcher instance was installed under /usr/sap/<CON>/SYS/exe/uc/os390, then you must adapt the two APL resources created by the HA-Wizard.

In the application resources SAP<CON>WD_STA and SAP<CON>WD_SRV, you must change and adapt the following path listed under USS Control:

```
/usr/sap/<CON>/SYS/exe/uc/os390_64/sapstart  
--> /usr/sap/<CON>/SYS/exe/uc/os390/sapstart
```

and

```
/usr/sap/<CON>/SYS/exe/uc/os390_64/sapstartsrv  
--> /usr/sap/<CON>/SYS/exe/uc/os390/sapstartsrv
```

- **The SAP Solution Manager Diagnostics Agent (SMDA) resources**

The SMDA instances on z/OS must be installed on every LPAR with their own SID. In the SA z/OS **SAPSRV add-on policy*, the sample SIDs of SM1, SM2, SM3 are used for a sample configuration with three SMDA installations. Those SIDs need to be adapted either by the SAP HA wizard or by a manual change to

the installed SMDA SIDs. Based on the sapstartsrv framework, three applications SAPSM<n>SM_SRV, SAPSM<n>SM, and SAPSM<n>SM_STA (n=1,2,3) are defined within two application groups (with SID SM1 shown in the example):

- SAPSM1SMD group
 - SAPSM1SM_STA which is the SMDA sapstart process
 - SAPSM1SM which is the SMDA Java main process
- SAPSM1SMSRV group
 - SAPSM1SM_SRV which is the sapstartsrv process

Figure 30 shows the class structure for Solution Manager Diagnostics Agent (SMDA) resources.

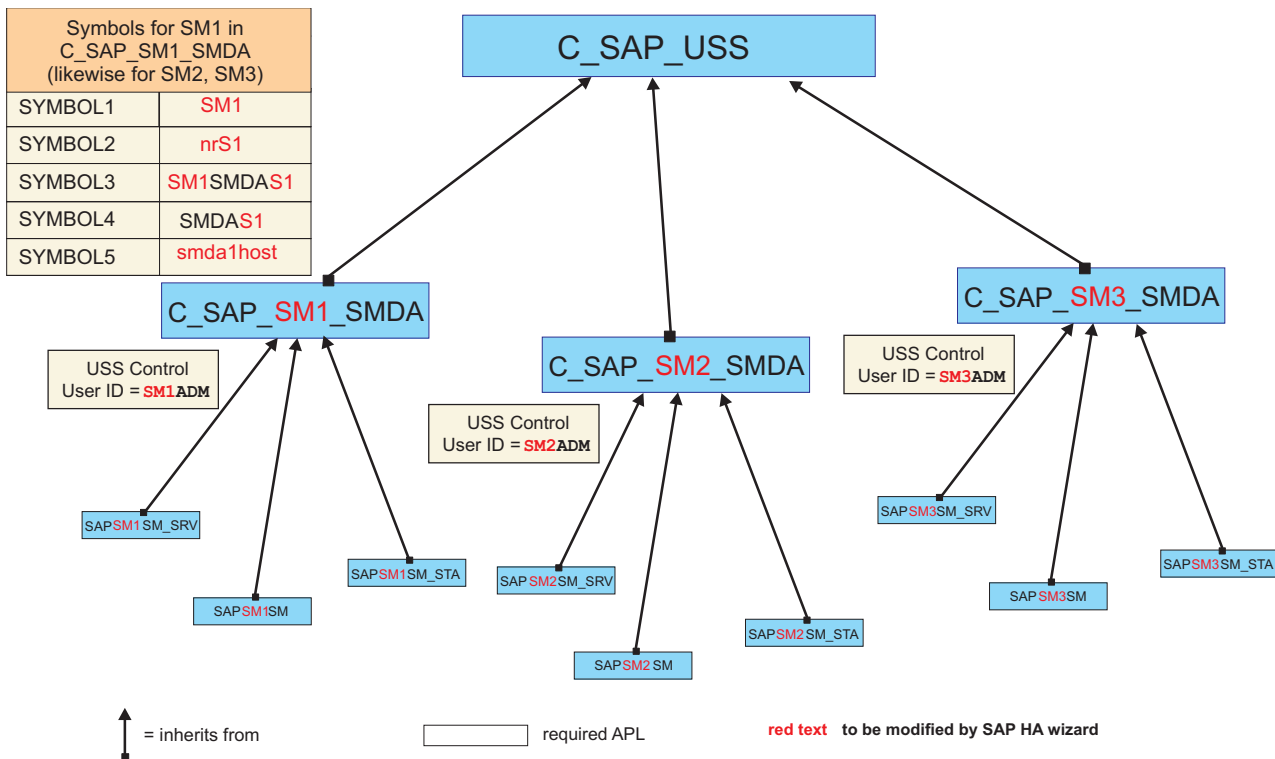


Figure 30. *SAPSRV add-on policy - class structure for Solution Manager Diagnostics Agent resources

SAP system-dependent groups

View this information to learn about the purpose and contents of the SAP system-dependent groups and their naming conventions.

Each SAP system-dependent group contains all the resources that are needed for the operation of a certain SAP system. The recommended general naming convention is SAP<SID>_X. Thus, you can easily filter on the NetView Management Console (NMC) or define SDF panels for monitoring the resources that are required by a specific SAP system.

Figure 31 on page 159 shows the details of the **SAPSID_X** group. This SID-specific group comprises the following components:

- ABAP SAP Central Services Group (SAP<SID>AENQX)

- ABAP enqueue and message server group (SAP<SID>ASCSX)
- ABAP enqueue replication server group (SAP<SID>AER_X)
- ABAP ERS sapstartsrv group (SAP<SID>ARSRVX [automation name S<SID>ARSRV_X])
- Java SAP Central Services Group (SAP<SID>JENQX)
 - Java enqueue and message server group (SAP<SID>JSCSX)
 - Java enqueue replication server group (SAP<SID>JER_X)
- Java ERS sapstartsrv group (SAP<SID>JRSRVX [automation name S<SID>JRSRV_X])
- DB2 database server group (DB2_X)

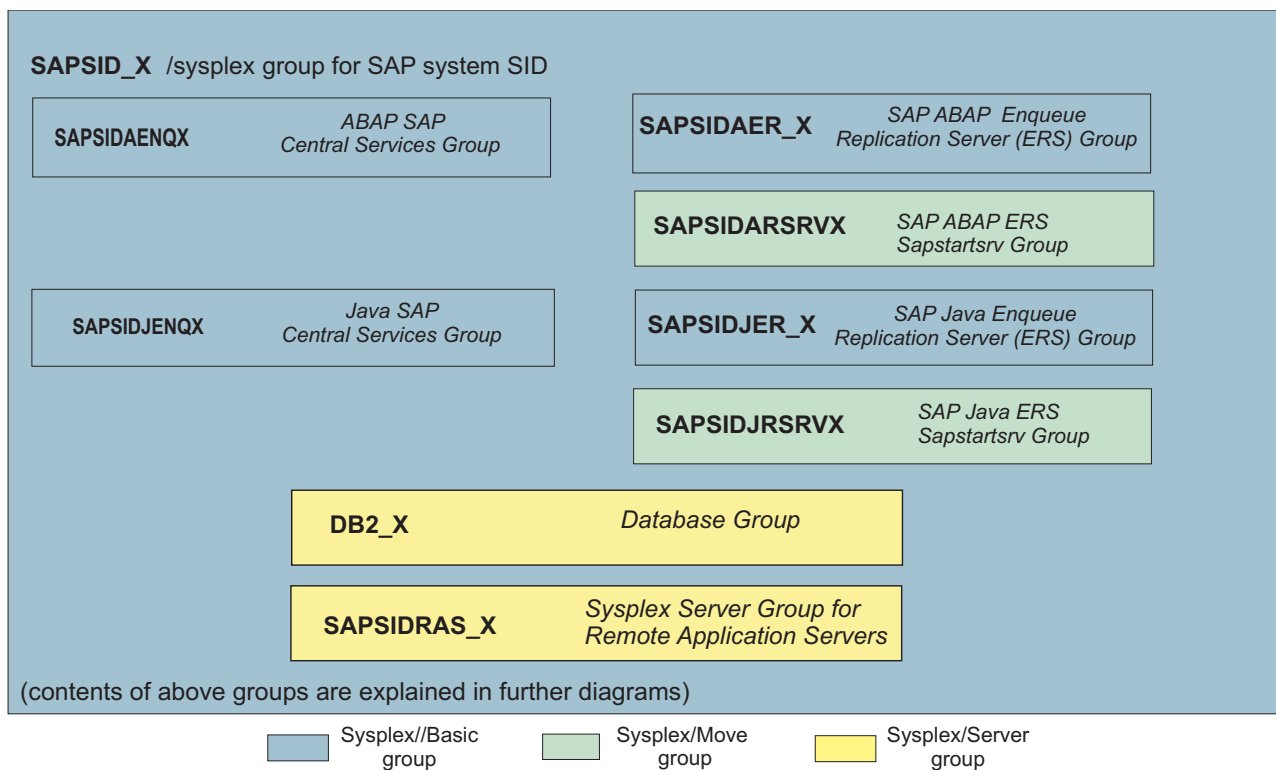


Figure 31. Group Sapsid_X belonging to SAP system SID

ABAP central services and enqueue replication server

Read information about the policy definition of the ABAP central services and the closely related enqueue replication server.

The mechanisms in the Java part (see “Java central services and enqueue replication server” on page 164) are similar to this one.

Figure 32 on page 160 shows the lowest level in the group structure of the ABAP central services and enqueue replication server.

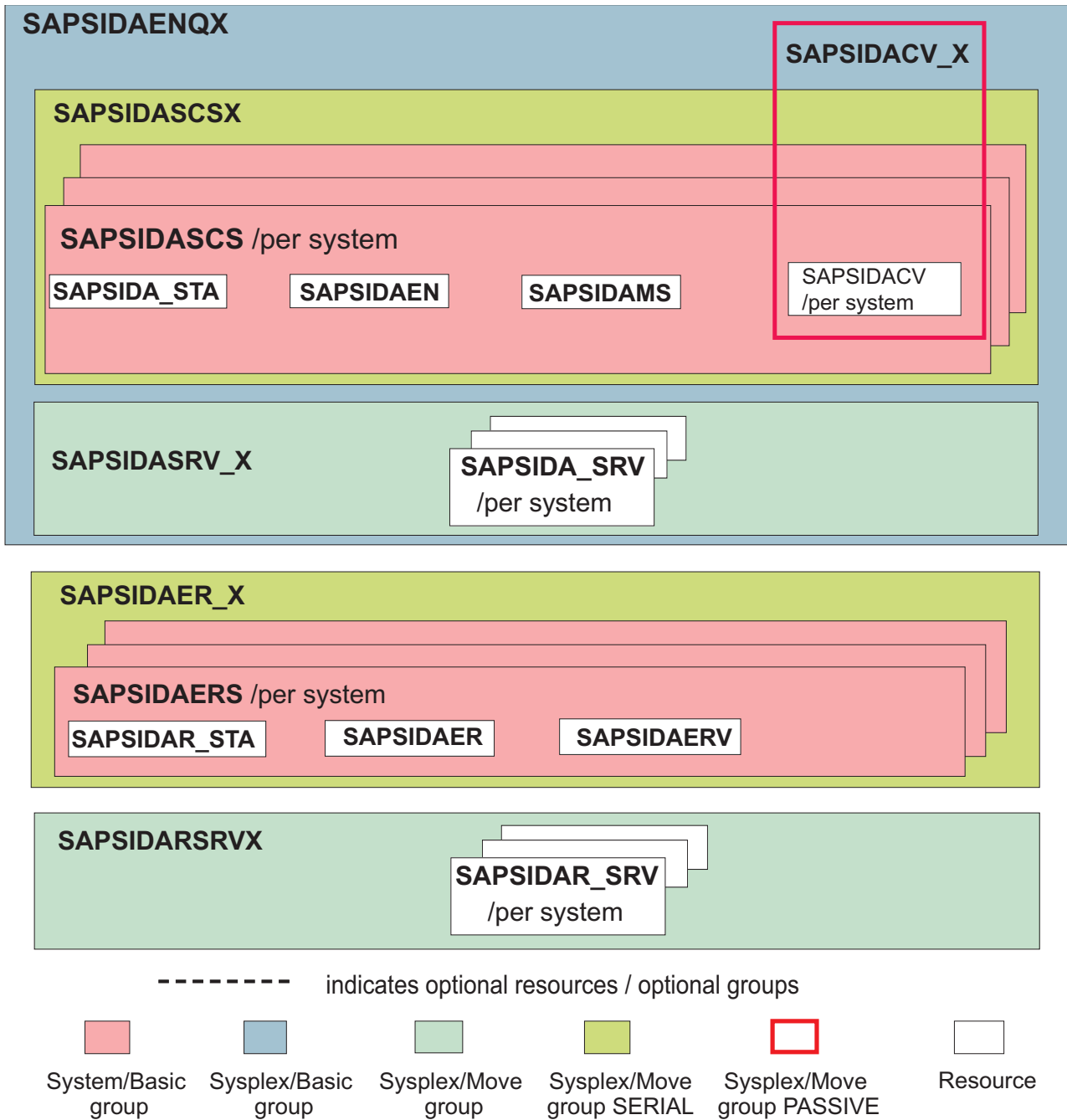


Figure 32. Lowest level in group structure of ABAP central services and enqueue replication server

SAP<SID>AENQX of Figure 32 contains these resources:

- the ASCS sapstartsrv process (SAP<SID>A_SRV), which must be active to start the ABAP SCS instance
- the VIPA associated with the ABAP SCS instance (SAP<SID>ACV)
- the ABAP sapstart process (SAP<SID>A_STA [subsystem name SAP<SID>AST]), which is the parent process of:
 - ABAP enqueue server (SAP<SID>AEN)
 - ABAP message server (SAP<SID>AMS)

The resources SAP<SID>AEN and SAP<SID>AMS have a *HasParent/StartsMeAndStopsMe* relationship to SAP<SID>A_STA.

SAP<SID>AER_X of Figure 32 on page 160 contains these resources:

- **SAP<SID>AR_SRV** [subsystem name SAP<SID>AR_SR], which models the ABAP ERS sapstartsrv process that must be active in order to start the ABAP ERS instance
- **SAP<SID>AERV**, which models the VIPA associated with the ABAP ERS instance
- **SAP<SID>AR_STA** [subsystem name SAP<SID>ARST], which models the ABAP ERS sapstart process that is the parent process of the ABAP enqueue replication server resource **SAP<SID>AER**.

SAP<SID>AER has an *Externally/StartsMe* relationship and a *MakeAvailable/WhenAvailable* relationship to **SAP<SID>AR_STA**.

Further details of Figure 32 on page 160:

- When the SAP system is configured with TCP/IP based replication with a separate enqueue replication server instance the *threshold definition* for the **SAP<SID>AEN** application is as follows:

Threshold	Value
Critical Number	1
Critical Interval	01:00:00
Frequent Number	1
Frequent Interval	02:00:00
Infrequent Number	1
Infrequent Interval	12:00:00

The critical threshold number of the enqueue server is set to **1**. This setting means that SA z/OS does not try to restart the enqueue server on the same LPAR. Instead, a failover to a different LPAR is triggered whenever the enqueue server terminates. When the SAP system is configured with EnqCF based replication (see Chapter 14) then a restart in-place of the enqueue server after a failure is possible without losing enqueue locks. Therefore, the threshold definition for the **SAP<SID>AEN** application should be as follows:

Threshold	Value
Critical Number	5
Critical Interval	00:10:00
Frequent Number	3
Frequent Interval	01:00:00
Infrequent Number	3
Infrequent Interval	12:00:00

- The VIPA resource **SAP<SID>ACV** (in addition to being member of the **SAP<SID>ASCS**) is also member of the **SYSPLEX MOVE PASSIVE** group **SAP<SID>ACV_X**. Its purpose is to define a relationship between the enqueue server its VIPA and the enqueue replication server. The relationship of the **SAP<SID>AER_X** group is *MakeAvailable/WhenAvailable(Passive/Weak)* to the **SAP<SID>ACV_X** group.

This ensures that the **INGGROUP** command (described in Table 16 on page 163) in the application automation definitions of the **SAP<SID>ACV** resource is processed by SA z/OS prior to the decision where to place the enqueue

replication server. Since SAP<SID>ACV_X is a MOVE group, only one VIPA is started or is active in the SYSPLEX at one time.

Additionally, the SAP<SID>A_STA resource has a *MakeAvailable/WhenAvailable(Active/Weak)* relationship to the VIPA. This relationship ensures that the VIPA is up before the ABAP SCS instance is started via the sapstart resource (SAP<SID>A_STA).

- The SAP enqueue replication server resource SAP<SID>AER is a member of the SAP<SID>AERS group, which in turn is a member of the SYSPLEX MOVE group SAP<SID>AER_X. This structure ensures that only one enqueue replication server is started or is active in the SYSPLEX at one time. For an explanation of the dependency and relationships between the enqueue server and the enqueue replication server, see “Dependencies between the ABAP enqueue server and the enqueue replication server.”
- Additionally, the SAP<SID>AER_X group has a *HasParent* relationship to SAP<SID>AR_SRV. This relationship ensures that the sapstartsrv service of the ABAP ERS instance is active and able to start/stop the ABAP ERS instance.

Dependencies between the ABAP enqueue server and the enqueue replication server

The SAP enqueue replication mechanism imposes certain restrictions on the location where the components run. This topic describes these dependencies and shows how to implement them.

The following considerations only apply to the case where traditional TCP/IP based replication is being used. When using *EnqCF replication* (see Chapter 13, “Enqueue replication into a z Systems coupling facility,” on page 277), there is no replication server.

- During normal operations, the enqueue server and the enqueue replication server (with its associated ERS VIPA) must not run on the same LPAR.
- Let us take an example where the enqueue server runs on LPAR A and the enqueue replication server runs on LPAR B. If the enqueue server fails, it must *not* be restarted on LPAR A:
 - Instead, the enqueue server must be restarted on LPAR B where the enqueue replication server is running.
 - Only in this case, the enqueue server can rebuild its enqueue table from the replicated copy of the enqueue table that was maintained by the enqueue replication server.
 - The enqueue replication server should now stop on LPAR B.
 - In order to re-establish high availability, the enqueue replication server should then be restarted on a different LPAR.

These mentioned restrictions are implemented in the SAP policy definition. Using the definitions within this policy, establish the following three dependencies between the enqueue server and the enqueue replication server:

- **Dependency 1:** The enqueue replication server is always started on a different LPAR from the one on which the enqueue server is running .
- **Dependency 2:** 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 .
- **Dependency 3:** Do not start the enqueue replication server on an LPAR where the enqueue server failed previously.

Note: The implementation of the dependencies is shown in subsequent sections for the *SAPSRV *add-on policy* in System Automation for z/OS 3.5. The previous implementation in SA z/OS 3.4 used the ACORESTART keyword which was replaced with the ANYSTART and REFRESHSTART mechanisms in SA z/OS 3.5.

Implementation of Dependency 1

The INGGROUP commands in the Messages and User data section of the policy definition for the SAP<SID>ACV resource ensure that the enqueue replication server is not started where the enqueue server (actually the related VIPA) is currently running. This is accomplished by setting the PREFER value to 1 for the enqueue replication server group (SAP<SID>AERS) and the LPAR where the VIPA for the enqueue server (SAP<SID>ACV) is running.

Table 16. Messages and User Data section from the SAPSIDACV policy definition

Message ID	Command Text
REFRESHSTART	INGGROUP SAP<SID>AER_X/APG,ACTION=ADJUST,MEMBERS=(SAP<SID>AERS/APG/&SYSNAME.), PREFER=(1)
RUNNING	INGGROUP SAP<SID>AER_X/APG,ACTION=RESET INGGROUP SAP<SID>AER_X/APG,ACTION=ADJUST,MEMBERS=(SAP<SID>AERS/APG/&SYSNAME.),PREFER=(1)

Implementation of Dependency 2

The INGGROUP commands in the Messages and User data section of the policy definition and in the “Startup” POSTSTART definitions of the SAP<SID>AER resource ensure that the enqueue replication server attracts the enqueue server if this fails. This is accomplished by setting the PREFER value to 700 for SAP<SID>ASCS and the LPAR where the enqueue replication server (SAP<SID>AER) is running. This requires, that the default PREFER value of each SAP<SID>ASCS group is set to 601 in the SAP<SID>ASCSX SYSPLEX MOVE group.

Table 17. Startup section from the SAPSIDAER policy definition

Message ID	Command Text
ANYSTART	INGGROUP SAP<SID>ASCSX/APG,ACTION=ADJUST,MEMBERS=SAP<SID>ASCS/APG/&SYSNAME.),PREFER=(700)
POSTSTART	INGGROUP SAP<SID>ASCSX/APG,ACTION=RESET

Implementation of Dependency 3

The *MakeAvailable/WhenObservedSoftDown* relationship against SAP<SID>ASCS/APG/= prevents the start of the enqueue replication server (SAP<SID>AER) whenever the ABAP central services group SAP<SID>ASCS on the same system is in HARDDOWN status.

Table 18. Relationships section from the SAPSIDAER policy definition

Relationship	Supporting Resource	Automation	Chaining	Condition
MakeAvailable	SAP<SID>ASCS/APG/=	PASSIVE	WEAK	WhenObservedSoftDown

Consequences of this implementation: A System Automation operator must manually change the status of the failed resource(s) in the group to AUTODOWN

(after he has investigated or resolved the root cause of the resource failure), in order to allow the enqueue replication server to start on that LPAR.

Alternative to this implementation: In a two-LPAR environment, the described mechanism prevents the enqueue replication server from being restarted by SA z/OS after a failure of the enqueue server. If you want to enable SA z/OS to do this restart you need to set a BROKEN enqueue server to AUTODOWN as soon as it is restarted on the other system. This can be done by the following policy change to the SAP<SID>AER definition:

1. Remove the *MakeAvailable/WhenObservedSoftDown* relationship to SAP<SID>ASCS/APG/=.
2. Add to the list of POSTSTART commands: SETSTATE SAP<SID>AEN,AUTODOWN.

One possible consequence of using System Automation to automatically reset the enqueue server (SAP<SID>AEN) status instead of letting an System Automation operator do it manually is that the enqueue server might start to move back and forth if the it fails over and over again with the same error:

1. Enqueue server fails on LPAR1.
2. SA z/OS moves it to LPAR2. There the enqueue server fails again.
3. SA z/OS then moves the enqueue server back to LPAR1, and so on.

You need to decide which is the best behavior for your installation and define the SAP<SID>AER resource accordingly.

Note: This outlined alternative solution to this implementation is not included in the *Best Practice Policy*.

Optional component of the ABAP central services

Learn why support for an optional ABAP gateway resource become obsolete.

The gateway service is not needed in the ASCS instance and is therefore not installed by the SAP central services installation. For more details see the SAP Netweaver 7.4 online documentation about the SAP gateway:

Gateway - SAP Library

Starting with SA z/OS 3.5 APAR level OA48922, the **SAPSRV add-on policy* is synchronized with the SAP SWPM default installation for ASCS. It no longer contains an ABAP gateway resource.

Refer to earlier versions of this publication (prior to SC33-8206-08) for details on how to model such an optional SAP gateway service.

Refer to earlier versions of this publication (prior to SC33-8206-08) for details on how to model such an optional SAP gateway service.

Java central services and enqueue replication server

This section describes the policy definition for the Java central services and the closely related enqueue replication server.

The same mechanisms and dependencies between the groups and resources apply as for the *ABAP central services* (described in “Dependencies between the ABAP enqueue server and the enqueue replication server” on page 162). Therefore, for

detailed explanations you should refer to the ABAP central services description (simply replacing the resource names in the explanation).

Figure 33 on page 166 shows the lowest level in the group structure of the Java central services and enqueue replication groups.

SAP<SID>JENQX contains these resources:

- The Java SCS sapstartsrv process (SAP<SID>J_SRV), which must be active in order to start the Java SCS instance
- The VIPA associated with the Java SCS instance (SAP<SID>JCV)
- Java sapstart process (SAP<SID>J_STA [subsystem name SAP<SID>JST]), which is the parent process of:
 - Java enqueue server (SAP<SID>JEN)
 - Java message server (SAP<SID>JMS)
 - Java gateway (SAP<SID>JGW) which was introduced with SAP 7.1

The three resources SAP<SID>JEN, SAP<SID>JMS and SAP<SID>JGW all have a *HasParent/StartsMeAndStopsMe* relationship to SAP<SID>J_STA.

SAP<SID>JER_X contains these resources:

- The Java ERS sapstartsrv process (SAP<SID>JR_SRV [subsystem name SAP<SID>JR_SR]), which must be active in order to start the Java ERS instance
- The VIPA associated with the Java ERS instance (SAP<SID>JERV)
- Java ERS sapstart process (SAP<SID>JR_STA [subsystem name SAP<SID>JRST]), which is the parent process of:
 - the Java enqueue replication server resource SAP<SID>JER.

The SAP<SID>JER has an *Externally/StartsMe* relationship and a *MakeAvailable/WhenAvailable* relationship to SAP<SID>JR_STA.

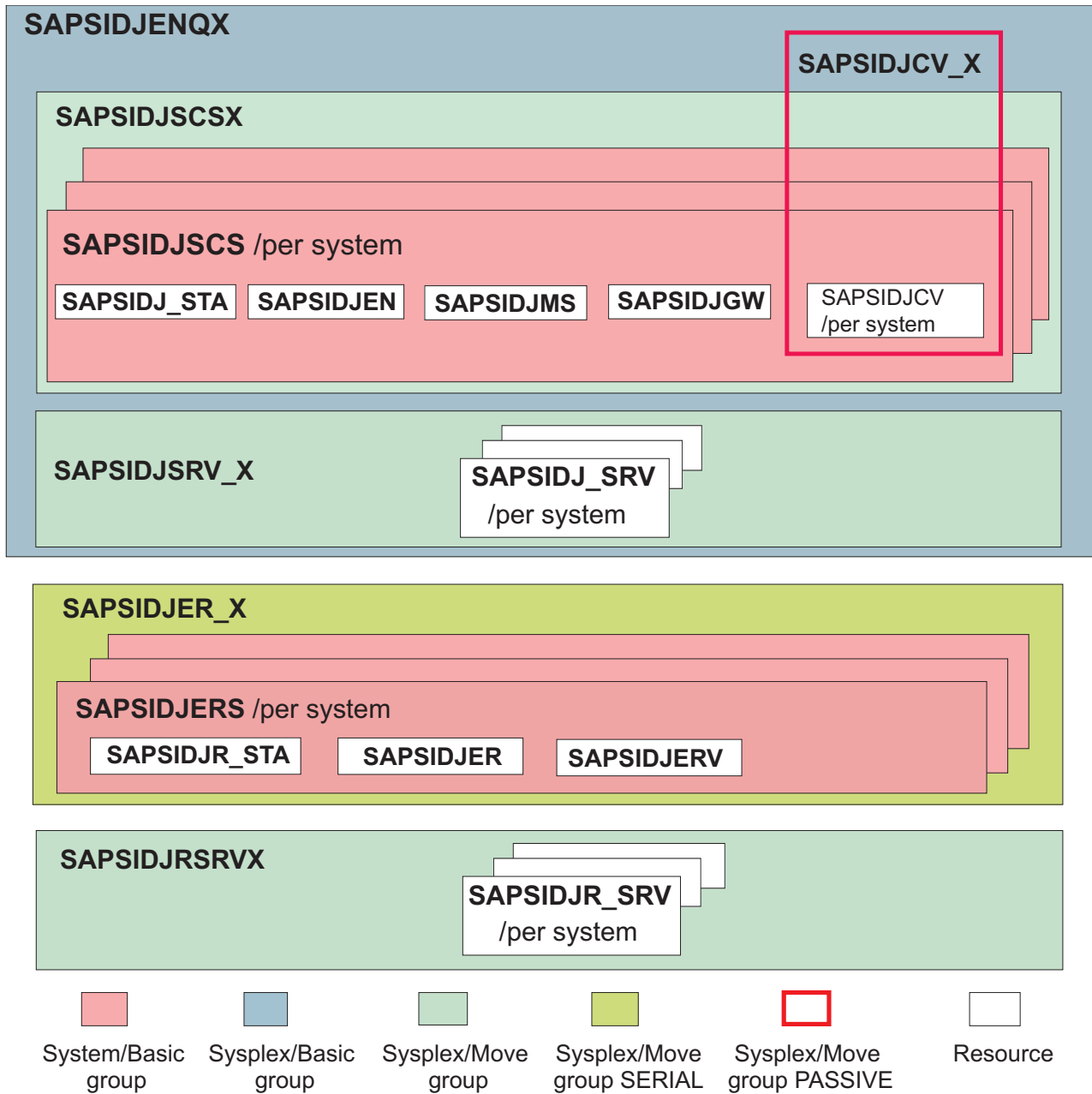


Figure 33. Lowest level in group structure of Java central services and enqueue replication groups

Monitoring the health state of SAP enqueue replication

SAP enqueue replication is essential for a high availability setup for SAP on z Systems - either with replication into the z/OS coupling facility, or running a separate enqueue replication server (ERS) instance. There are events, when enqueue replication can be suspended, for example, during peak workload phases. Such an event is only reported and indicated by SAP in the *enquelog* trace file, which can be found in the work directory of your SAP Central Services instance, or in the *dev_enqrepl* developer trace file.

If a failover of the enqueue server happens in such a situation (when replication is suspended), all enqueue lock data is lost. Normally, if enqueue replication becomes deactivated, then it is automatically activated again. Reasons for suspension might be temporary problems, like:

- disruptions of the TCP/IP or XCF communication between the enqueue server and the replication server
- temporary buffer shortages on the enqueue server host that is caused by a workload spike.

With 7.40 kernel, the enqueue server has been enhanced to write a *replication started* and *replication stopped* message to the z/OS system console, if instructed by a new profile variable. See *SAP Note 1899862* for details. The NetView automation table is enhanced to trap both messages and to set the health status of the System Automation monitor resource associated with the enqueue server resource. Therefore, System Automation can notice and remember intervals and events with suspended replication.

Prerequisites:

- **Prerequisites from SAP:** Refer to *SAP Note 1899862*.

You must run the enqueue server with 7.40 kernel and patchlevel 31 or later. You must switch on the reporting of messages via the SAP profile parameter `enque/server/system_console = true`

and restarted the enqueue server.

- **Prerequisites from System Automation:** The **SAPSRV add-on policy* shipped with SA z/OS 3.5 contains the following resources and definitions to enable health state monitoring of the SAP enqueue replication:

- the monitor resource `SAP<SID>AEN_M` with a *PeerOf* relationship to `SAP<SID>AEN` for ABAP enqueue server (not shown in Figure 32 on page 160)
- the monitor resource `SAP<SID>JEN_M` with a *PeerOf* relationship to `SAP<SID>JEN` for Java enqueue server (not shown in Figure 33 on page 166)
- the *HasMonitor* relationship in `SAP<SID>AEN` and `SAP<SID>JEN`
- an entry in the automation table for catching the enqueue server messages.

Health state monitoring for SAP enqueue replication runs only if you observe the “Naming conventions” on page 91. If you do not adhere to these conventions, you need to specify your own rules in the NetView automation table. For this purpose, in the `INGMSG01` entry of this table, copy the entries for *SAP Enq.Replication started or stopped* and adapt them according to your conventions.

Operational details:

If the monitor is `AVAILABLE`, but its health state is `N/A`, this can mean:

- The enqueue server started OK on LPAR X, but the ERS does not start on any other LPAR. Such a situation might occur:
 - if the SAP profile parameter `enque/server/system_console` was not set to true, or the SAP Kernel level is lower than 7.40
 - if the ERS resource has one or more `StopVotes`
 - if the ERS start is inhibited, because of an earlier failure of the enqueue server, or of any other resource of the SAP Central Services group (see “Implementation of Dependency 3” on page 163).

- The enqueue server started OK on LPAR X, but does not get a connection to the CF.

Check those conditions and resolve the problems.

The monitor is AVAILABLE and the health state is **Normal**, if the enqueue server started OK, and replication is active. In this case, the following message is written to the z/OS console:

```
BPXF024I (<SID>ADM) SAP system <SID> instance <ASCS/SCS> enqueue replication started
```

The monitor is AVAILABLE and the health status is **Warning**, if the enqueue server started OK, and replication was active and now becomes suspended. In this case, the following message is written to the z/OS console:

```
BPXF024I (<SID>ADM) SAP system <SID> instance <ASCS/SCS> enqueue replication stopped
```

If replication resumes (starts again), the **Warning** health status is switched back to **Normal**.

Alternatively it is possible to generate an alert.

SAP HA Interface for SA z/OS

The SA z/OS product provides an implementation of the SAP HA Interface. With this interface installed and enabled, SAP management components can work seamlessly together with SA z/OS while the SA z/OS policy is actively managing the SAP central services on z/OS. This topic provides an overview of the benefits and functions of the SAP HA Interface.

Information about the complete setup and installation is delivered with the SA z/OS product documentation in *SAP HA interface for System Automation for z/OS - Installation and Setup* (PDF file HA1ibSAzOS.pdf), which describes:

- the necessary steps to install and activate the SAP HA Interface for SAP systems where the SAP central services are automated via System Automation for z/OS
- the resulting behavior of the SAP system with the SAP HA Interface enabled.

For all current SAP NetWeaver 7.x releases, SAP instances are started and stopped through the SAP service program `sapstartsrv`. This program runs similar to a UNIX daemon on every server where SAP instances are to be started or stopped. It offers a web service interface that accepts a number of different requests. SAP administrators or SAP tools use the `sapcontrol` program to send requests to `sapstartsrv` that trigger a start or stop of an SAP instance.

Benefits of the SAP HA Interface

In an environment where automation software is used to manage (start and stop) the SAP SCS instances, such an external invocation of `sapcontrol` to stop or start an SAP instance might be in conflict with the goals of the automation software. For example, if an SAP instance on z/OS is stopped by an external SAP administrator command, an active System Automation policy immediately restarts the instance.

However, an SAP system that is controlled by automation software should tolerate external starting and stopping of SAP instances in the following scenarios:

- SAP maintenance tools like the *Software Update Manager* are not aware that an automation software is active. They should be able to stop and (re)start SAP SCS instances by using `sapcontrol` as needed.

- SAP administrators should be able to use the *SAP Management Console* (SAP MC) for SAP maintenance operations, or use `sapcontrol` commands from the command line.

For this purpose, SAP defined an HA Interface which enables `sapstartsrv` to interact with the automation software. In an SAP system that has the HA library activated, both the automation software and an external request can stop or start SAP instances without interfering with each other. The SAP HA Interface can either be implemented as a shared library, or as a script, and it is supplied by the vendor of the automation software. See also *SAP Note 1693245: SAP HA Script Connector Library*.

The SAP HA Interface on z/OS is primarily intended for use with SAP Central Services instances. With SA z/OS APAR OA48922 or higher, the SAP HA Interface is in addition enabled for use with SAP Web Dispatcher and SAP Solution Manager Diagnostics Agent instances. The SAP HA Interface on z/OS is implemented via:

- the SAP script connector shared library `saphascriptco.so` that is shipped with the SAP kernel for z/OS Unix System Services
- the cluster connector REXX script `sap_os390_cluster_connector.rex` that is shipped with the System Automation for z/OS (SA z/OS) product from IBM

For SAP application servers that are modeled as remote resources in SA z/OS, the HA interface is implemented via:

- the SAP script connector shared library `saphascriptco.so` (for Linux) or `saphascriptco.o` (for AIX), which are shipped with the SAP kernel on these platforms
- the cluster connector shell script `sap_os390_cluster_connector.sh` that is shipped with the System Automation for z/OS product from IBM.

SAP defined extensions to its HA interface - the SAP HA Interface Version 2. For details see the *SAP Note 1822055: Enhanced SAPHA library interface*. These extensions are supported by versions 2.0 or higher of `sap_os390_cluster_connector.rex` and `sap_os390_cluster_connector.sh`, and require SA z/OS 3.5.

SAP HA Interface shipment and documentation for SA z/OS

The implementation of the SAP HA Interface for SA z/OS is contained in the cluster connector scripts `sap_os390_cluster_connector.rex` and `sap_os390_cluster_connector.sh` that are delivered with the System Automation for z/OS 3.5 product in `/usr/lpp/ing/SAP/ING_sap.tar`.

The documentation for this implementation is provided in PDF file *HALibSAzOS.pdf* that is also contained in the mentioned tar file.

DB2 policy

Learn how the **SAPSRV add-on policy* and the **DB2 add-on policy* work together.

In its SAP<SID>_X group, the SA z/OS **SAPSRV add-on policy* contains a reference to a DB2_X group. The definition of this group is in fact contained in the SA z/OS **DB2 add-on policy*. The SAP HA wizard adapts the resources contained in the SA z/OS **DB2 add-on policy* to the actual SAP system by replacing

occurrences of the *DB2* string in the resource definitions with the 3-character SID of your actual SAP system, if the **DB2 add-on policy* policy is part of the wizard's source PDB.

The **DB2 add-on policy* in SA z/OS 3.5 contains additional DB2xADMT resources for the *DB2 administrative task scheduler*. Since this address space is in most cases not required for SAP, it is not shown in Figure 34. If you do not require the *administrative task scheduler*, you should remove the ADMT resources from the HA wizard's source PDB before generating a new SAP policy.

Figure 34 shows an example of what the DB2 part of the policy looks like if your SID is **HA1** and for a data sharing system with three members.

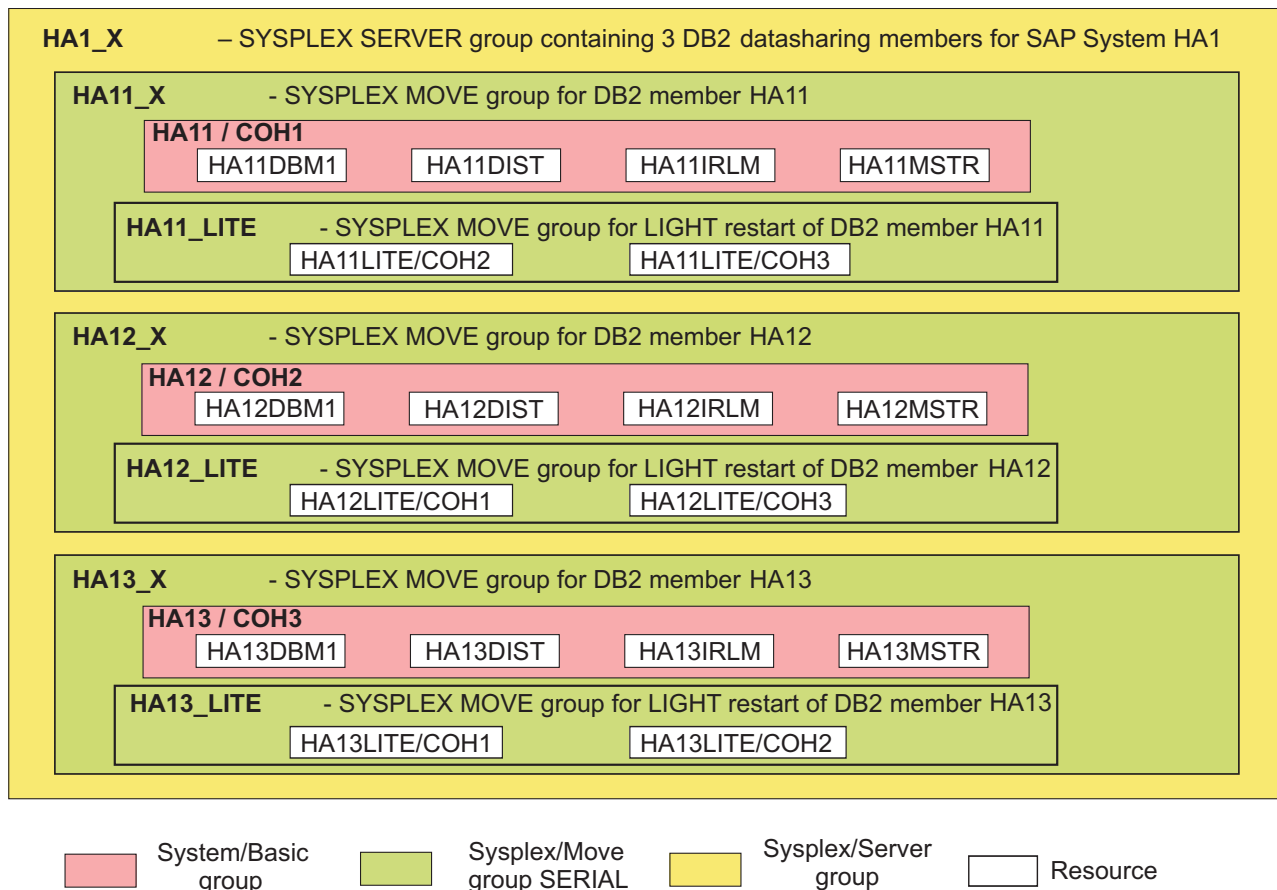


Figure 34. DB2 **Best Practice Policy** – adapted for SAP system HA1

In the sample shown in Figure 34, the following three DB2 members are defined for the sample SAP system **HA1**:

- HA11 running on LPAR COH1
- HA12 running on LPAR COH2
- HA13 running on LPAR COH3

Each of these members is represented by a SYSPLEX/MOVE named HA1<n>_X. This in turn contains a SYSTEM/BASIC group which again contains the standard DB2 address space resources as members.

The failure of a DB2 member requires a **LIGHT restart** of that member on another LPAR, such that DB2 is able to clean up its locks. This LIGHT restart capability is

implemented in the **DB2 add-on policy* in form of the SYSPLEX/MOVE groups HA1<n>_LITE_X. For a more detailed explanation of the relationships mechanisms inside the **DB2 add-on policy*, refer to the SA z/OS documentation.

Note: The **DB2 add-on policy* that is shipped with SA z/OS 3.5 uses a value of CSONLY for the **Status Determination** field in the HA1<n>_X and the HA1<n>_X groups definitions. Therefore the COMPOUND status of the HA1_X group then shows the worst COMPOUND status of any contained member resource, making it easier to detect problems with any of the DB2 address spaces. If your active SA policy was generated using an SA z/OS 3.4 version of the **DB2 add-on policy* you may want to add the CSONLY value.

You may consider using a value of CSONLY for the **Status Determination** field in the HA1_X and the HA1<n>_X groups definitions. The COMPOUND status of the HA1_X group then shows the worst compound status of any contained member resource, making it easier to detect problems with any of the DB2 address spaces.

Note: The SA z/OS **DB2 add-on policy* that is shipped with SA z/OS 3.5 has CSONLY set for these groups.

The **DB2 add-on policy* models the DB2 data sharing group as a SYSPLEX SERVER group with availability target *ALL. Consequently, the group goes into DEGRADED status if one of the DB2 member could not be started. If you model your SAP application servers as proxy resources as described in “SAP application servers as proxy resources,” then these proxy resources have a *MakeAvailable/WhenAvailable* relationship to the DB2 sysplex group. Therefore, the application server resources is not started if the DB2 group is in DEGRADED status. If you configure DB2 connection failover for your application servers (see “DB2 connection failover” on page 15), then it may be sufficient that **one** DB2 data-sharing member is available - allowing all application servers to be started. In order to enable this in the SA z/OS policy, you may consider changing the satisfactory target of your DB2 SERVER group from *ALL to a different value which is smaller than the number of DB2 data sharing members, for example 1.

If you do not use the SAP HA wizard, you must adapt the DB2 group and resource names to your own SAP system and your environment. However, you should also take account of the naming conventions described in “Naming conventions” on page 91.

Note: The SAP add-on policies shipped with SA z/OS contain a reference to the DB2_X group. If you do not use the SAP HA wizard to generate the policy for your SAP system, you must ensure that your top-level DB2 group is correctly referenced and included in the SAP group SAP<SID>_X.

SAP application servers as proxy resources

In the high availability Option 1 described in this publication, the ABAP/Java application servers are controlled via so-called *proxy resources*. The subtopics of this information unit describe how these different types of application servers can be modeled as such proxy resources in SA z/OS.

High availability Option 2 (see “Option 1: Automation using System Automation for z/OS only” on page 24) relies on System Automation for Multiplatforms to control SAP application servers.

| This section describes the setup for “Option 1: Automation using System
| Automation for z/OS only” on page 24 where the application servers are remotely
managed by SA z/OS.

The modeling of remote SAP application server resources described here is contained in the **SAPSRV add-on policy* and SA z/OS 3.5, or with SA z/OS 3.4 at a minimum APAR level OA43145. SAP Application Servers for dual-stack systems are modeled as System/Server group which keeps the dual-stack resource available in cases where one stack resource is not or no longer operational.

Depending on the SAP solution, your SAP system application servers can be:

- ABAP-only,
- Java-only, or
- dual-stack (ABAP plus Java).

Creating a remote application server policy

This topic enables you to create a remote application server policy. It informs about a starting point and the prerequisites for this task and provides an overview of the relationships for remote application servers between elements of the **SAPSRV add-on policy*.

As a starting point for creating an SA z/OS policy for remote application servers, you should use the SA z/OS customization dialog to import the remote application server resources from the **SAPSRV add-on policy* into your new policy database. Make sure that you have SA z/OS 3.5 installed, or that your SA z/OS 3.4 installation includes at least APAR level OA43145.

All sample remote application server resource definitions are contained under the Sysplex Server group SAP<SID>RAS_X. as outlined in figure Figure 35 on page 173. After importing the remote application server resources into your policy database, you should see all the resources groups and applications as shown in Figure 35 on page 173.

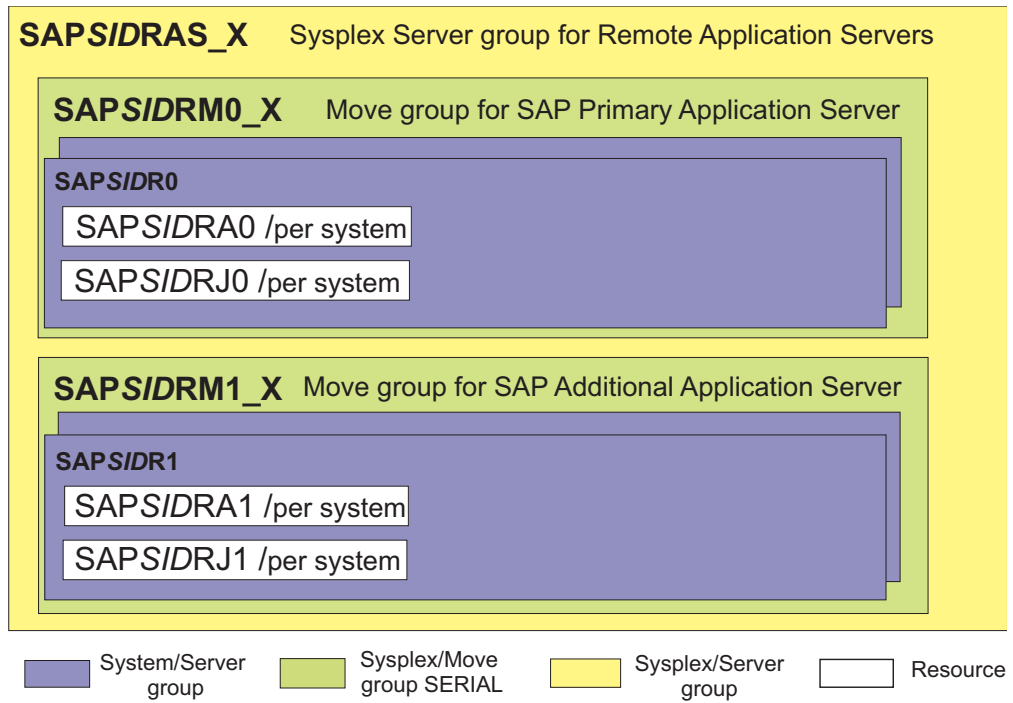


Figure 35. Server group for remote application servers

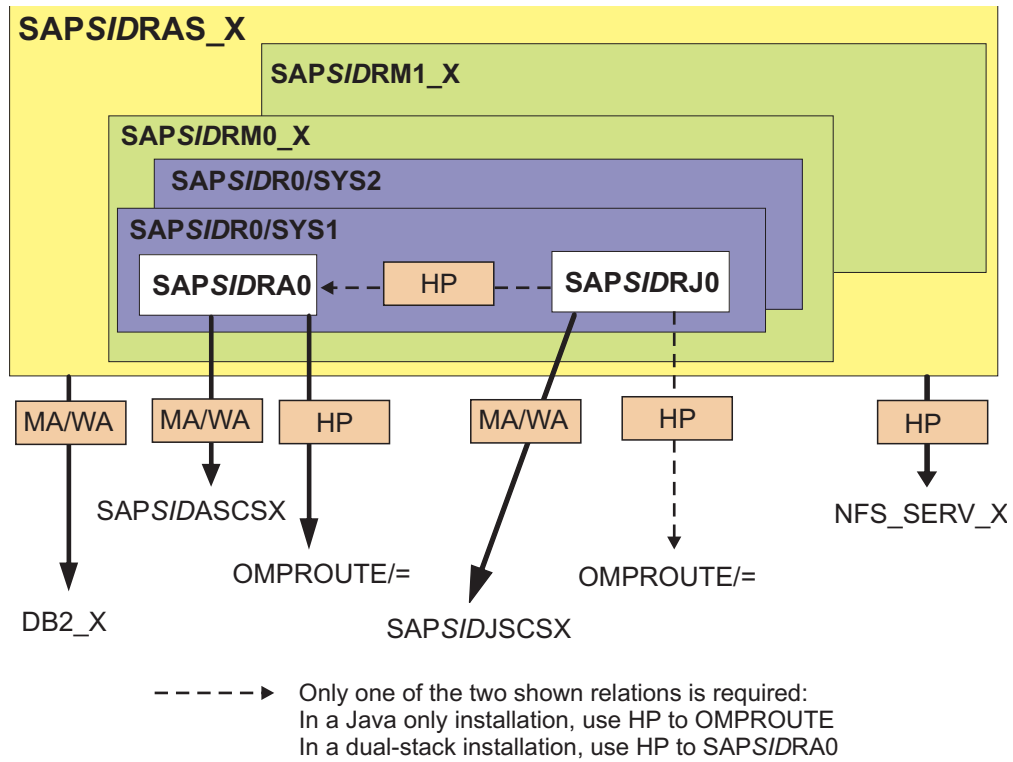


Figure 36. Overview of the relationships for remote application servers between elements of the *SAPSRV add-on policy

Figure 36 shows the existing relationships of the remote application server resources. Relationships can define a dependency together with a condition

(MA/WA stands for *MakeAvailable/WhenAvailable*) or without a condition (HP stands for *HasParent*). For a discussion of the implications of the relationship between the remote application server resources and DB2, refer to the information provided in topic “SAP application server groups” on page 178.

Scripts for the remote application server policy

The SA z/OS policy for remote application servers relies on several scripts for its operation. Learn how to obtain and where to install these scripts, and read their detailed descriptions.

These scripts are contained in the `ING_sap.tar` file that is shipped with the SA z/OS product. They must be placed into the directory which is referenced by the SA z/OS remote application server resources. So before activating the remote application server policy, you need to copy the scripts to this directory. If you place the scripts into a different directory, you need to modify the command strings in the STARTUP and SHUTDOWN sections of the proxy resources `SAP<SID>Rxx` in your policy accordingly.

Note: The **SAPSRV add-on policy* assumes that the scripts are placed in the home directory of the SAP administrator user: `/u/<sid>adm`

The remote application servers are managed by the following scripts:

```
start_as
check_as
stop_as
samsapctrl_asping
```

For detailed information see “Automation scripts” on page 285.

Details and configuration for application servers as proxy resources

This information describes how to manually adapt the proxy resources that are shipped with the **SAPSRV add-on policy*. The SAP HA wizard provided with SSA z/OS 3.5, or with SA z/OS 3.4 at a minimum level of APAR OA43145, is able to automate most of these steps.

For details see the file `SAPHAWizard.pdf`, which is shipped inside the `ING_sap.tar` file in the SA z/OS z/OS UNIX directory `/usr/lpp/ing/SAP`.

SAP dual-stack (ABAP + Java) systems require an additional application group level in the SA z/OS policy to combine the two pieces, ABAP and Java. The **SAPSRV add-on policy* includes this group level to support SAP dual-stack systems. For ABAP only systems and for Java only systems, some resources are not required and should be removed:

- For an ABAP-only system, remove all System/Server application groups `SAP<SID>R0`, `SAP<SID>R1`, and the Java application resources `SAP<SID>RJx`.
- For a Java-only system, remove all System/Server application groups `SAP<SID>R0`, `SAP<SID>R1`. and the ABAP application resources `SAP<SID>RAx`.

For dual-stack (ABAP+Java) resources you need to make specific adaptations to the resources which are outlined in “Dual-stack (ABAP plus Java) application server” on page 178.

Common changes for all SAP application server types

The **SAPSRV add-on policy* and the description in this topic contain sample resources and groups with names containing a placeholder of <SID>. You need to replace it in all resources and group names with your own SAP system ID of the system for which you are creating the new policy.

Make sure you also replace all occurrences of the placeholder <SID> inside the resource definitions (for example, in the APPLICATION INFO: the STARTUP and SHUTDOWN commands and in the USS Control specification).

For example: **SAP<SID>RAS_X** should be changed to **SAPHA1RAS_X** if **HA1** is your SAP system ID.

If you want to automate more than the two remote application servers that come with the **SAPSRV add-on policy*, then you need to generate new resources by copying from existing resources and groups observing the naming conventions. The SAP HA wizard documentation describes a procedure that uses the INGEIMP sample job of SA z/OS to add remote application server resources to your policy. See the file SAPHAwizard.pdf, which is shipped inside the ING_sap.tar file in the SA z/OS z/OS UNIX directory /usr/lpp/ing/SAP. If you want to automate less than the two remote application servers that come with the **SAPSRV add-on policy*, then you need to remove the ones that are not needed from your policy.

ABAP-only application servers

The application SAP<SID>RA0 is a z/OS UNIX proxy resource which is used to automate a primary ABAP application server running on a remote system, for example under AIX, Linux on z Systems, Linux on System x, or Windows.

Additional ABAP application servers are modeled as SA z/OS resources SAP<SID>RA1, SAP<SID>RA2, and so on. The customization steps for SAP<SID>RA0 described in this section need to be executed for all ABAP application server resources.

Because the application server runs on a remote system, it cannot be directly started, stopped, or monitored by SA z/OS using native z/OS commands. The start, stop, and check scripts (see “Scripts for the remote application server policy” on page 174), issue the actual SAP instance start or stop commands via SSH remote command execution. An application server status indication is provided by the monitor routine of the check script.

Java-only application servers

The application SAP<SID>RJ0 is a z/OS UNIX proxy resource which models a primary JAVA application server running on a remote system, for example under AIX, Linux on z Systems, Linux on System x, or Windows.

Additional JAVA application servers are modeled as SA z/OS resources SAP<SID>RJ1, SAP<SID>RJ2, and so on. The customization steps described in this section need to be executed for all Java application server resources.

Similar to ABAP-only application servers, a Java application server instance cannot be directly started, stopped, or monitored by SA z/OS using native z/OS commands. The start, stop, and check scripts (see “Scripts for the remote

application server policy” on page 174) issue the actual SAP instance start or stop commands via SSH remote command execution. An application server status indication is provided by the monitor routine of the check script.

In RELATIONSHIPS, change the *HasParent* relation from the ABAP resource SAP<SID>RA<n> to OMPROUTE as the supporting resource. For example: OMPROUTE/APL/=.

Policy configuration for ABAP-only and Java-only application servers

The STARTUP, SHUTDOWN, and POSTSTART sections and the z/OS UNIX Control specification in each remote application server resource definition need to be adapted to the specific SAP application server instance of your SAP system. The SAP HA wizard will make these adaptations, if it has access to the SAP profiles for these SAP instances.

If you need to adapt the resource definition manually, you should observe that most of the parameters in the command lines of STARTUP, SHUTDOWN, and POSTSTART are supplied via System Automation SYMBOLS - so you need to modify the SYMBOL values. The sample screen shot from Figure 37 on page 177 shows the SYMBOL definitions for the SAP<SID>RA0 resource as it is delivered with the **SAPSRV add-on policy*.

General parameters

General parameters apply to all remote application server resources of a given SAP system. The SYMBOL definitions (SYMBOL1 and SYMBOL2) for these parameters are contained in the class **C_SAP_<SID>RAS** for ABAP application server instances and **C_SAP_<SID>RJS** for Java application server instances. Their SYMBOL definitions for these 2 symbols are inherited by each application server resource. Figure 37 on page 177 shows these in **bold font**. When adapting these SYMBOLS, you should do this in the CLASS definition. In the definition for SYMBOL1 you need to replace **<SID>** with the 3-character SAP system ID of your system.

In most cases you can leave SYMBOL2 as it is. If you make changes in SYMBOL2, then consider the following:

- The string SSH must not be changed, since it is the only supported communication mechanism.
- For ABAP application server resources, the value **12** in SYMBOL2 stands for the number of retries (with a 10 second wait time between retries), that the startup script start_as waits for successful startup of the application server. If it takes longer than $12 \times 10 = 120$ seconds to start the ABAP application server in your environment, you should increase this counter. For a Java application server resource, the default number of retries in SYMBOL2 is **24**.

Important: You might need to adapt the number of retries for a failover situation as described hereafter, if the VIPAs of the DB server under z/OS belong to a sub-net which is forward-able by the default gateway as described in “Additional considerations” on page 53. If the LPAR of the DB member to which a connection should be established is not up, then the failover to an alternative member takes much longer. To cover this, you can for example, double this value.

Application server specific parameters

The symbols SYMBOLS3 through SYMBOL6 that are shown in *italic font* in Figure 37 are specific to an application server instance.

- SYMBOL3 must contain the name of the SAP instance, for example, DVEBMGS12
- SYMBOL4 must contain the host name of the SAP application server that is used to start the instance.
- In the two job names - SYMBOL5 and SYMBOL6 - you only need to substitute the substring **<SID>** with the 3-character SAP system ID of your system.

```
Application Symbols
Entry Type : Application          PolicyDB Name : SAP35TMP
Entry Name : SAPSIDRA0          Enterprise Name : SAP35TMP
SYMBOL1 . . . . . SID
Description 1 . . . . . 3-character SAP System id for remote SAP app servers
SYMBOL2 . . . . . 12 SSH
Description 2 . . . . . number of retries and method for starting remote SAP ABAP AS
SYMBOL3 . . . . . RAS_A00_INST
Description 3 . . . . . Instance name for SAP remote app server instance
SYMBOL4 . . . . . RAS_A00_HOST
Description 4 . . . . . Host name of SAP remote app server
SYMBOL5 . . . . . SIDA0STA
Description 5 . . . . . Job name for commands during app server startup
SYMBOL6 . . . . . SIDCAROC
Description 6 . . . . . Job name to be used for the prestart copy commands
SYMBOL7 . . . . .
Description 7 . . . . .
SYMBOL8 . . . . .
Description 8 . . . . .
SYMBOL9 . . . . .
Description 9 . . . . .
```

Figure 37. Application Symbols

For both ABAP and Java remote application server resources, two STOP commands are defined in the policy:

- the **SHUTNORM** command, which kills only the monitor routine. When the monitor routine has shut down, the remote application server appears to be down for SA z/OS. After a move of the resource to a different LPAR, the new monitor routine reconnects 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.
- the **SHUTFORCE** command, which actually sends a stop command to the SAP application server.

A new option is available with SA z/OS 3.5 APAR level OA48922, which eliminates for SA z/OS operators the explicit usage of the **SHUTFORCE** command in order to really stop the application server itself. If the proxy resource definition uses the new REXX script SAPRASTP, then this script decides automatically when to use **SHUTNORM** or **SHUTFORCE**. The decision is based on the stop votes existing in SA z/OS for the proxy resource:

- If a stop vote with desired state UNAVAILABLE for the move group exists, then the remote Application Server is stopped.
- Otherwise only the proxy resource is stopped and automatically restarted on a different LPAR. This suggests that the stop was for an LPAR maintenance scenario.

In order to exploit this option, the **SHUTNORM** policy must be changed. Replace as follows:

Cmd Ps	AutoFn/*	Command Text
Pass 1:	INGUSS	/bin/kill -2 &SUBSPID ==> with SAPRASTP 2
Pass 4:	INGUSS	/bin/kill -9 &SUBSPID ==> with SAPRASTP 9

The REXX program SAPRASTP is delivered with SA z/OS 3.5 APAR level OA48922 and higher as part of the `ING_sap.tar` file in the z/OS UNIX directory `/usr/lpp/ing/SAP`. Add the SAPRASTP REXX program to a user-defined data set within the NetView command list, for example `USER.SINGNREX`. These data sets are specified by the **DSICLD DD** statement of NetView.

Important: The **SAPSRV add-on policy* of SA z/OS 3.5 APAR level OA48922 automatically exploits this option. Therefore, you must add the SAPRASTP REXX program to the NetView command list.

Dual-stack (ABAP plus Java) application server

In an SAP dual-stack application server installation, there is physically one SAP instance which runs both the ABAP and the Java stack.

Within that instance, both stacks are started by default when the (ABAP) instance starts. The SA z/OS policy needs to split the single instance into two logical parts and defines two different resources:

- An ABAP application server with a remote ABAP application server resource. In order to eliminate the explicit usage of the STOP FORCE mode for an SA z/OS operator, the SHUTNORM policy of the ABAP application server resource must be changed. Replace the kill commands with SAPRASTP as explained in “Application server specific parameters” on page 177.
- A Java application server with a remote JAVA application server resource

However, there is a close relationship between those two logical application servers:

- The Java instance starts after the ABAP instance is active.
- *Stopping* the Java application server does *not* stop the Java server processes. It only stops the monitoring program `samsapctrl_asping`, which does a basic health check of the Java application server.

The definitions for the *logical* ABAP application server part of a *dual-stack* instance are those for an ABAP-only application server (listed in “ABAP-only application servers” on page 175). For the primary (ABAP) application server instance this is resource `SAP<SID>RA0`.

You must manually apply the following change in the corresponding JAVA resource `SAP<SID>RJ0` such that both ABAP and JAVA resources do actually model the behavior of a dual-stack application server:

- In the SHUTDOWN processing, remove the FORCE step.
Repeat this action for all resource definitions for dual-stack application server resources `SAP<SID>RA1/SAP<SID>RJ1`, and so on.

SAP application server groups

This section explains the resource group definitions. For additional information, also refer to Figure 35 on page 173.

The remote application server resources from the **SAPSRV add-on policy* include the following SA z/OS application groups for a single SAP system. These groups allow you to easily monitor and manage the remote application server resources via SA z/OS.

SAP<SID>RAS_X

This SERVER group contains all MOVE groups SAP<SID>n_X as members.

With the help of a SERVER group, it is possible to bring down SAP application servers for maintenance without SA z/OS triggering any action. The server group will only show a degraded group status.

You might consider setting an availability target (AVT) of the SERVER group to N-1 for your N remote application servers. In this case the group status will be OK, even if one application server is not available. As another option you might set the satisfactory target (STGT) to a number even less than this to avoid an error status of the SERVER group as long as the number of active application servers is equal to or higher than the STGT.

The default for both the availability target and the satisfactory target is *ALL.

Starting with SA z/OS 3.4 APAR level OA43145, the **SAPSRV add-on policy* contains an explicit *MakeAvailable/WhenAvailable* relationship between the remote SAP application server and DB2. This means, an SAP application server can start as soon as one DB2 member is up. However, there is no check whether the definitions for DB2 connection failover (in connect.ini, db2dsdriver.cfg, or config.xml configuration files) are consistent with these SA z/OS definitions. This must be ensured manually.

The SAP<SID>RAS_X group has an *HasParent* relationship to the NFS server sysplex group NFS_SERV_X/APG, because the application server executables reside on the NFS. If the NFS server is moved, the application servers do not need not be stopped since such a move is transparent to the NFS clients. Before the NFS server is stopped completely and not available on any system, the application servers must be stopped. Before the application servers can be started, the NFS Server on z/OS must be running.

SAP<SID>RMn_X

In case of a single stack SAP installation, ABAP or Java, each of these MOVE groups (n=0,1,2) is defined to include resources for one remote SAP application server. The recommendation is to define SAP<SID>RA0 for the primary SAP ABAP application server resource and SAP<SID>RA1, SAP<SID>RA2, ... and so on, for additional SAP ABAP application servers of the SAP system <SID>. For SAP Java application servers it is SAP<SID>RJ0, SAP<SID>RJ1, ... and so on.

In case of a dual-stack SAP installation, each MOVE group includes a System/Server group SAP<SID>R0, SAP<SID>R1, ... and so on. Each of this System/Server groups then includes resources for one remote SAP application server pair (ABAP & Java). The recommendation is to define SAP<SID>RA0 for the Primary SAP ABAP application server resource and SAP<SID>RA1, SAP<SID>RA2, ... for additional SAP ABAP application servers of the SAP system SID. For SAP Java application servers, it is SAP<SID>RJ0, SAP<SID>RJ1, ... and so on.

With the definition of a MOVE group, each remote application server can be monitored via SA z/OS from one LPAR. If this LPAR is stopped, the monitoring of the remote application server is moved to another LPAR in the sysplex via the MOVE group. You need a resource definition for each LPAR that should host the monitoring task. See Figure 35 on page 173 where active SA applications are represented as white boxes. When an LPAR is stopped, the application servers themselves are not stopped. Refer to Chapter 12, "Operating an SAP system under System Automation control," on page 243 for maintenance scenarios.

Groups, applications, and relationships

This topic provides a graphical overall view of groups and applications contained in the **SAPSRV add-on policy*.

Figure 38 on page 181 presents an overall view of all the SA z/OS groups that are contained in the **SAPSRV add-on policy*.

To improve readability, SYSTEM/Basic groups are only shown as *one box*, although *one such group per LPAR* exists in the SYSPLEX.

Figure 39 on page 182 provides an overview of the relationships of the **SAPSRV add-on policy*.

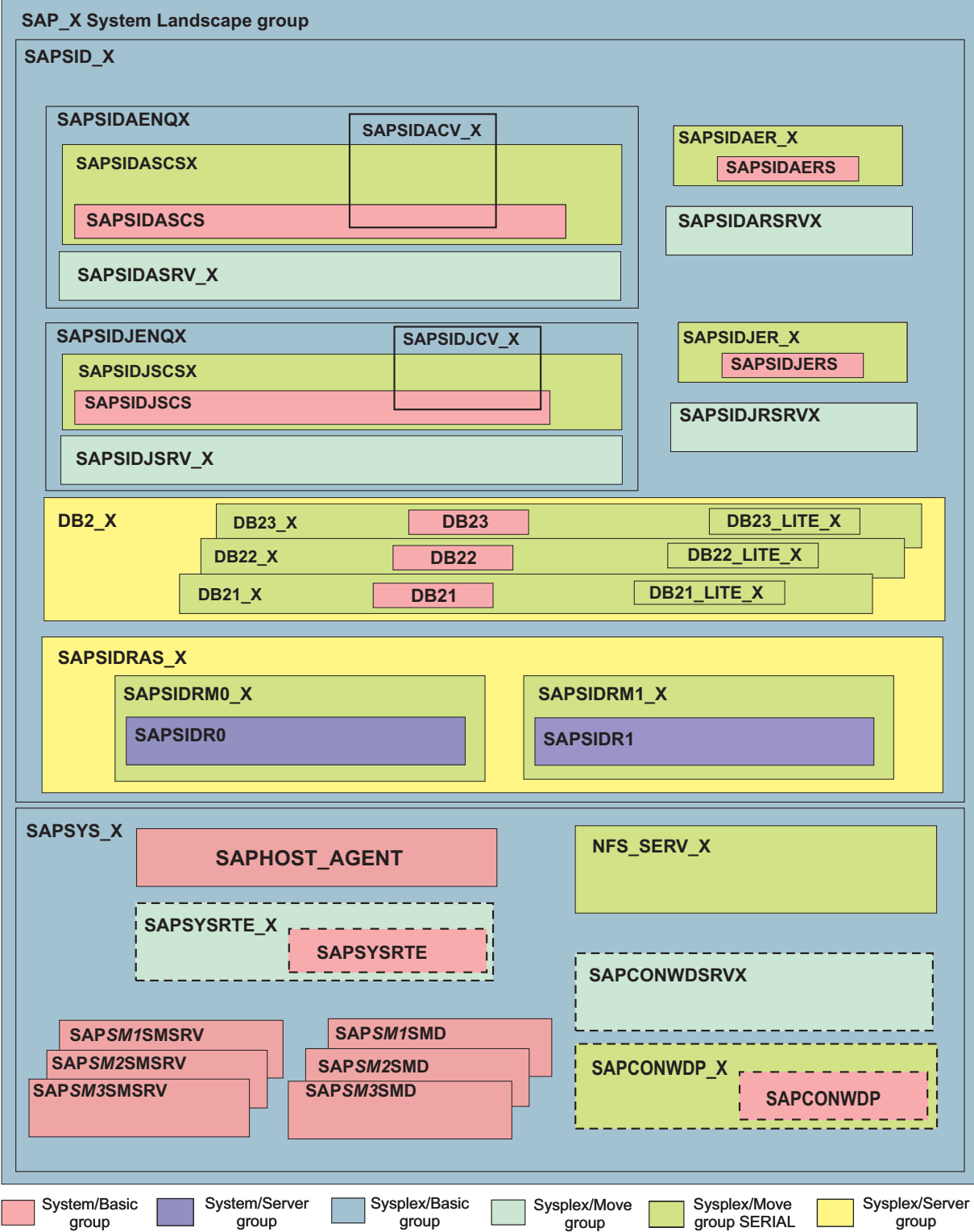


Figure 38. Overview of the resource groups

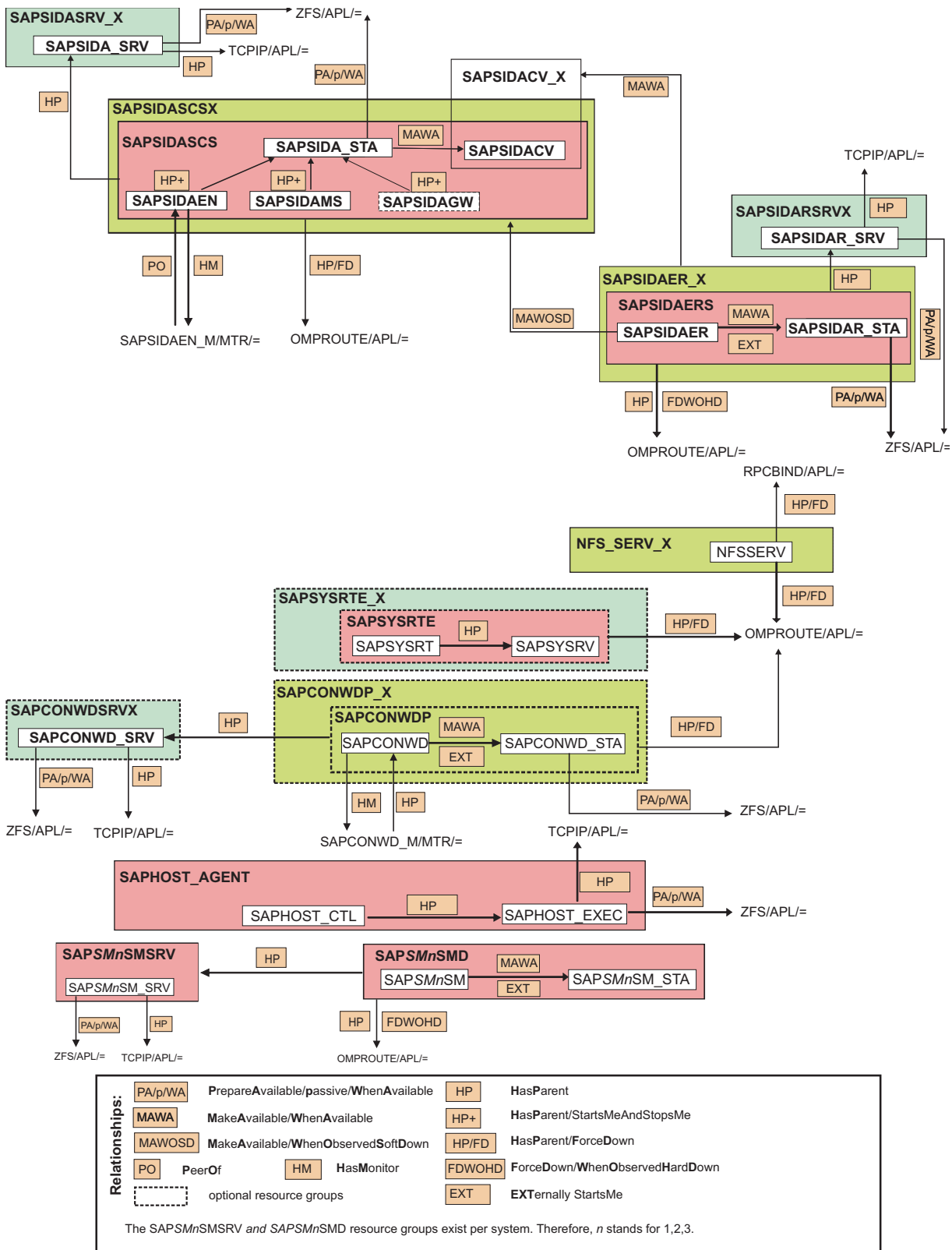


Figure 39. Overview of the relationships between elements of the *SAPSRV add-on policy (excluding DB2 elements)

Notes to Figure 39:

- The figure shows the ABAP resources as an example. The differences to Java resources are the following:

- The resource names containing the pattern <SID>A would be named with the pattern <SID>J instead.
- The SAP<SID>JGW resource would be mandatory.
- The figure does not show those resources that do not have any dependencies on other resources, like for example the VIPA of the Web Dispatcher.
- The *MakeAvailable/WhenAvailable* relationship between (ABAP or Java) enqueue replication server and the VIPA for the (ABAP or Java) enqueue server is needed to ensure that the enqueue replication server is started after the enqueue server and its VIPA have started.
- The *MakeAvailable/WhenObservedDown* relationships between (ABAP or Java) enqueue replication server and the (ABAP or Java) enqueue server group implements the dependency between them. These are listed in the description of **Dependency 3** in “Dependencies between the ABAP enqueue server and the enqueue replication server” on page 162.

Chapter 9. Customizing System Automation for Multiplatforms

The topic describes the implementation and design of the automated and highly available SAP system that is driven by the base component of Tivoli System Automation for Multiplatforms. It provides guidance and recommendations for a high availability strategy regarding SAP environments. It also discusses practical considerations regarding design and implementation.

This information unit contains these main subtopics:

- “System Automation for Multiplatforms for SAP on IBM z Systems”
- “System Automation for Multiplatforms with HA Option 2”
- “Platform-specific recommendations” on page 187

System Automation for Multiplatforms for SAP on IBM z Systems

SAP on IBM z Systems is built around IBMDB2 for z/OS as the SAP database server. The application logic, which is written in ABAP or Java, is supported on several platforms. This section covers the operating systems Linux on z Systems and AIX and the implementation of an automated and highly available SAP system partly or completely based on the Tivoli System Automation for Multiplatforms product.

Note: Although Windows is a valid application server platform for SAP on IBM z Systems, it is not supported by System Automation for Multiplatforms (SA MP) 4.1 or later.

The SAP resources that you need to automate and make highly available with System Automation for Multiplatforms depend on the option that you select for your SAP high availability environment. The available options are described in Chapter 3, “Planning for high availability for SAP,” on page 21. SA MP is part of the configuration that is described as *Option 2*.

System Automation for Multiplatforms with HA Option 2

With HA Option 2, you use System Automation for Multiplatforms (SA MP) to automate the components that are listed in this topic on AIX or Linux operating systems.

These components automated with the help of SA MP on AIX or Linux are the following:

- the SAP Central Services (for ABAP or for Java, or both),
- the SAP application server (AS) instances,
- the SAP utilities with an affinity to those application servers like SAProuter or SAP Web Dispatcher,
- the NFS server (if you decided to run the NFS server within the SAP cluster for just this SAP system). As an alternative you might also run NFS server in its own SA MP cluster serving several SAP systems.

The SAP database (DB2 for z/OS) resides on z/OS and should be kept highly available with SA z/OS.

How to implement the SA MP policy for SAP

Read the contained information about the advantages of using the *SAP high availability policy feature* to make each component of an SAP system highly available.

Previous versions of this documentation included instructions how to implement an SA MP policy for SAP based on the as-is sample SAP HA policy that was contained in the zSAP_BusinessContinuity.zip file.

This as-is sample policy contained resource definitions for the SAP components for HA Option 2 (see Chapter 3, "Planning for high availability for SAP," on page 21). A drawback of this sample policy was the fact that it was not based on and did not take into account the SAP sapstartsrv service. Therefore, seamless integration of an SA MP controlled SAP Central Services instance into SAP systems management tools, like SAP's Management Console, or into SAP lifecycle tools, like Software Update Manager (SUM) was not possible.

The System Automation for Multiplatforms product provides a feature to make each component of an SAP system highly available. This *SAP high availability policy feature* is available for some time for SAP installations based entirely on AIX or Linux - including the SAP database.

With System Automation for Multiplatforms 4.1.0.2 or higher, the *SAP high availability policy feature* is now able to support SAP systems based on DB2 for z/OS. In particular, it can be used to automate all SAP components that are required for HA Option 2. The feature supports and models the SAP sapstartsrv service. Therefore, seamless integration of an SA MP controlled SAP Central Services instance into SAP systems management tools, like SAP's Management Console, or into SAP lifecycle tools, like Software Update Manager (SUM) is possible. SA MP policies for SAP can be configured via a policy wizard and the generated policy is supported by the SA MP product.

Recommendation

New installations or, for example, when redefining the SA MP policy of an existing SAP system, it is recommended that you switch to the *SAP high availability policy feature* of SA MP. For existing SAP installations, you can continue to use any SA MP policy that is based on the as-is SAP policy.

Using the SAP high availability policy feature to generate a policy

With System Automation for Multiplatforms 4.1.0.2 or higher, it is recommended that you use the *SAP high availability policy feature* that is provided with the SA MP product.

The *SAP high availability policy feature* supports the generation of an SA MP policy for Option 2. For details on installing and activating this feature in SA MP, see *Tivoli System Automation for Multiplatforms Version 4 Release 1 High Availability Policies Guide (SC34-2660-02)*, which is available from the IBM Knowledge Center for *Tivoli System Automation for Multiplatforms*.

The System Automation for Multiplatforms includes a policy wizard, which helps you to interactively create an SA MP policy for a SAP system.

The policy wizard is invoked with the command **sampolicy -w**, together with a policy template file. A description of the policy wizard and the SAP policy templates is contained in topic *SAP Central Services high availability policy in Tivoli System Automation for Multiplatforms Version 4 Release 1 High Availability Policies Guide (SC34-2660-02)*, which is available on IBM Knowledge Center for Tivoli System Automation for Multiplatforms.

To generate an SA MP policy for **HA option 2**:

- Use the wizard with the template files `sap_ABAP_v41.tmp1.xml`, `sap_JAVA_v41.tmp1.xml`, or `sap_DoubleStack_v41.tmp1.xml`, depending on the type of your SAP system, which is either an ABAP-only, Java-only, or ABAP + Java dual-stack (double-stack) system.
- Use the wizard with the template file `sap_AppServer_only_v41.tmp1.xml`, if you want to automate application servers with System Automation for Multiplatforms and if you use SA z/OS to control SAP Central Services on z/OS.

Using the sample SAP HA policy to generate a policy

Read this information, if you want to implement an SA MP policy for SAP based on a sample policy that is contained in previous versions of the `zSAP_BusinessContinuity.zip` file

Previous versions of this documentation included instructions how to implement an SA MP policy for SAP based on the as-is sample SAP HA policy that was contained in the `zSAP_BusinessContinuity.zip` file. See Edition 12/2013 or earlier of *Business Continuity for SAP on IBM z Systems*.

Platform-specific recommendations

This information discusses the following topics:

- “CPUPLUGD utility and Tivoli System Automation for Multiplatforms under Linux on z Systems”
- “RSCT network performance considerations” on page 188
- “Special considerations for running System Automation for Multiplatforms under virtual operating systems or LPARs” on page 188

CPUPLUGD utility and Tivoli System Automation for Multiplatforms under Linux on z Systems

If you run System Automation for Multiplatforms in a Linux on z Systems environment together with the `cpuplugd` utility, you must adapt the `cpuplugd` update interval in such a way, that `cpuplugd` does *not* contribute to application monitor command timeouts such as:

```
Feb 11 17:25:01 ih1scoh1 GblResRM[4800]: (Recorded using libct_ffdc.a cv 2)
:::Error ID: :::Reference ID: :::Template ID: 0:::Details File: :::Location:
RSCT,Application.C,1.92.1.16,3189 :::GBLRESRM_MONITOR_TIMEOUT
IBM.Application monitor command timed out.
Resource name SAPECC_ABAP_HA1_ih1scoh1_DVEBMGS02_AS
```

If Tivoli System Automation for Multiplatforms detects an *application monitor command timeout*, the corresponding resource is put into UNKNOWN status. As long as one such resource status exists, Tivoli System Automation for Multiplatforms does *not* perform any automation tasks, that is, no failover, no restart, and so on.

It is recommended to determine the default `MonitorCommandTimeout` (MCT) of all resources and then either:

- A. Set the update interval to a smaller value than the smallest MCT
- B. Increase the MCT for the resources if possible
- Adapt your environment with a combination of A and B

RSCT network performance considerations

When running System Automation for Multiplatforms on Linux under z/VM, it is recommended that you avoid the RSCT network ping heartbeat.

For more information, refer to *Tivoli System Automation for Multiplatforms: Administrator's and User's Guide*.

Special considerations for running System Automation for Multiplatforms under virtual operating systems or LPARs

This information describes the rules and considerations that are related to using *quorum nodes* if you are running a System Automation for Multiplatforms cluster under a virtual operating system or LPAR.

Examples include:

- z/Linux under z/VM and over multiple CECs
- z/Linux LPARs and over multiple CECs
- AIX LPARs and over multiple physical machines

By default, System Automation for Multiplatforms allows each node to be a quorum node, and expects only a single node to fail at any point in time. This is because System Automation for Multiplatforms considers a node to be a self-sufficient single physical host/system. However:

- A Linux on z Systems guest depends on its z/VM host.
- The z/VM host depends on the underlying z Systems LPAR and CEC.
- An AIX LPAR depends on the underlying physical machine, and so on.

By running several virtualized System Automation for Multiplatforms cluster nodes under one virtual operating system (OS) or physical machine, an outage (planned or unplanned) of that virtual OS or physical machine causes the outage of all those "hosted" virtualized System Automation for Multiplatforms cluster nodes at the same time.

A System Automation for Multiplatforms quorum is required for System Automation for Multiplatforms to function in the expected automated manner. During the outage of a virtual OS or physical machine, the resulting loss of several virtualized nodes can easily result in System Automation for Multiplatforms being unable to establish the required quorum. In such a case, no automation would take place on the remaining System Automation for Multiplatforms nodes, and if those nodes have active critical resources they are rebooted by System Automation for Multiplatforms. It is not consistent with the aim of using System Automation for Multiplatforms to establish a highly available environment.

For example, consider a System Automation for Multiplatforms cluster with three Linux on z Systems nodes (`lnxsapg`, `lnxsaph`, and `lnxsapi`), where `lnxsapg` and `lnxsaph` run under z/VM on CEC1, and `lnxsapi` runs under another z/VM on

CEC2. Each node is a quorum node by System Automation for Multiplatforms default. An outage of CEC1 results in an outage of its z/VM and in turn in an outage of both lnxsapg and lnxsaph.

As most of nodes is down, lnxsapi no longer has the operational quorum, and System Automation for Multiplatforms restarts node lnxsapi, if this node runs critical resources.

One solution to the problem that is described in the previous paragraph is, for example, to remove the quorum of lnxsaph. Only lnxsapg on CEC1 and lnxsapi on CEC2 are quorum nodes. It does have two consequences:

- You need a tiebreaker, as the cluster now has an equal number of quorum nodes.
- In case of an outage of CEC1 the node lnxsapi gets the operational quorum and takes over floating resources that were running on CEC1 nodes.

Such a change in the distribution of quorum nodes within the System Automation for Multiplatforms cluster additionally means:

- If the non-quorum node lnxsaph is down for whatever reason, for example, maintenance, then nothing happens to lnxsapg or lnxsapi, as both quorum nodes are active.
- If a quorum node, for example, lnxsapg, is down for whatever reason, for example maintenance, this has no impact on both other nodes as long as the other quorum node lnxsapi gets the quorum via the tiebreaker and the non-quorum node lnxsaph is able to communicate with lnxsapi.
- If, for example, your staff leave the quorum node lnxsapg down because they do not know which node is a quorum node, lnxsaph is stopped if the other quorum node becomes inactive.

Therefore, it is recommended that you have simple and symmetric cluster configurations.

You should distribute virtual System Automation for Multiplatforms cluster nodes equally over physical machines. If you have two physical machines (CECs), run one virtual node on each machine. You have a two node cluster. Or run two virtual nodes on each physical machine. You have a four node cluster. If you have three physical machines (CECs), run one virtual node on each machine. You have a three node cluster. Or run two virtual nodes on each physical machine. You have a six node cluster. These examples give you an impression of what a simple and symmetric cluster setup is. You must set up a tiebreaker for equal quorum nodes clusters.

If this is not possible, the more general and complex rule applies: Make as many cluster nodes as possible into quorum nodes, and at the same time ensure that any single failure of a physical system would not lose quorum. Two scenarios are possible. The following sample uses System Automation for Multiplatforms cluster setups with Linux on z Systems nodes under z/VM.

Even number of machines (CECs)

This information provides considerations that are related to using *quorum nodes* in an installation with an even number of machines.

According to the general rule described in “Special considerations for running System Automation for Multiplatforms under virtual operating systems or LPARs” on page 188, the number of quorum nodes within the cluster must be even, and

any single failure (CEC, z/VM LPAR or Linux on z Systems LPAR) must not result in the loss of more than 50% of the quorum nodes. Consider the following example:

```

CEC1
-----
zVM1   | zVM2   | CEC2
-----|-----|-----
- zLinux1 | - zLinux6 | - zLinux10
- zLinux2 | - zLinux7 | - zLinux11
- zLinux3 | - zLinux8 | - zLinux12
- zLinux4 | - zLinux9 | - zLinux13
- zLinux5 |          |

```

This is a System Automation for Multiplatforms cluster, which consists of a total of 13 nodes. The number of quorum nodes must be divided evenly over the two CECs, although zVM3 on CEC2 can only support a maximum of four quorum nodes. Therefore, there should be four quorum nodes on CEC1, and those four nodes must be evenly divided between zVM1 and zVM2.

The recommended configuration in this case is:

```

zVM1 = 2 Quorum Nodes
zVM2 = 2 Quorum Nodes
zVM3 = 4 Quorum Nodes

```

Odd number of machines (CECs)

This information provides considerations that are related to using *quorum nodes* in an installation with an odd number of machines.

With an odd number of CECs your setup must consider: If any *one* of them fails, then the remaining two CECs must still hold the quorum. In other words, the number of quorum nodes on any CEC must not exceed the number of quorum nodes on the remaining CECs. For example:

```

CEC1           | CEC2           | CEC3
-----|-----|-----
zVM1   | zVM2   | zVM3   | zVM4
-----|-----|-----
- zLinux1 | - zLinux6 | - zLinux10 | - zLinux14
- zLinux2 | - zLinux7 | - zLinux11 | - zLinux15
- zLinux3 | - zLinux8 | - zLinux12 | - zLinux16
- zLinux4 | - zLinux9 | - zLinux13 |
- zLinux5 |          |

```

In this example, the maximum number of lost cluster nodes occurs if CEC1 fails, and that is nine. Therefore, CEC2 and CEC3 requires a combined total of nine nodes, but they have seven. So for this setup, it is recommended that you reduce the number of nodes to a maximum of seven nodes on CEC1.

A possible recommendation for the shown sample configuration is as follows:

```

zVM1 = 4 Quorum Nodes
zVM2 = 3 Quorum Nodes
zVM3 = 4 Quorum Nodes
zVM4 = 3 Quorum Nodes

```

You can set quorum nodes by using the following command for each host name:

```
chrsrc -s 'Name == "hostname"' IBM.PeerNode IsQuorumNode=1
```

You can set non-quorum nodes by using the following command for each host name:

```
chrsrc -s 'Name == "hostname"' IBM.PeerNode IsQuorumNode=0
```


You can verify the state of each node by using the following command example:

```
ihlsco2:~ # lsrnnode -QB
Name      OpState  RSCTVersion  Quorum  Tiebreaker
ihlsco3   Online   2.5.1.2      No      No
ihlsco1   Online   2.5.1.2      Yes     Yes
ihlsco2   Online   2.5.1.2      Yes     Yes
```

In the shown example, ihlsco1 was running on one z Systems CEC1 and ihlsco2-3 on another z Systems CEC2.

When a System Automation for Multiplatforms cluster splits into two subclusters each with 50% of the quorum nodes, a tie breaker is required to determine if the subcluster should continue or not. System Automation for Multiplatforms (RSCT) is unable to determine if the reason for the cluster-split is due to dead cluster nodes or network problems prevent communications with the other nodes.

The probability that a network problem exists, reduces with the:

- Number of network interfaces that are used
- Redundancy of network equipment used

If you use a simple single network between all of your nodes, you must use System Automation for Multiplatforms tie breaker DASD, which must be shared between all of the Guests across all z/VM LPARs and z Systems CECs.

- The Reserve/Release DASD commands are used to establish the SAP MP operational quorum subcluster.
- This tie breaker DASD should be defined in the SYSTEM CONFIG or each z/VM system as SHARED to allow RESERVE/RELEASE DASD commands to be issued.
- There must be a LINK statement with options MW to allow multiple write access in the USER DIRECTORY of each Linux Guest.

Issue the following commands to make System Automation for Multiplatforms use the tie breaker DASD *nnnn*:

```
mkrsrc IBM.TieBreaker Name=DasdTieBreaker Type=ECKD
DeviceInfo="ID=nnnn" HeartbeatPeriod=5

chrsrc -c IBM.PeerNode OpQuorumTieBreaker="DasdTieBreaker"
```

To verify whether the tie breaker is being used, issue the following command:

```
lsrsrc IBM.TieBreaker
```

For more recommendations on using ECKD™ tie breaker under z/VM see the **IBM Knowledge Center - System Automation for Multiplatforms**. The Release Notes contain the latest updates.

If you have multiple redundant networks between all of your nodes, you must use *System Automation for Multiplatforms Network tie breaker*. This is because the failure of multiple redundant networks is so unlikely that the cause of System Automation for Multiplatforms (RSCT) not being able to communicate with another node is most likely that the node is dead. In such a case the overhead of a DASD tie breaker is unnecessary, and a simple network ping to an external gateway suffices to ensure that your own node is operational.

The following commands need to be issued to make System Automation for Multiplatforms use the network tie breaker x.x.x.x:

```
mkrsrc IBM.TieBreaker Type="EXEC" Name="NetworkTieBreaker"  
DeviceInfo='PATHNAME=/usr/sbin/rsct/bin/samtb_net  
Address=x.x.x.x Log=1' PostReserveWaitTime=30
```

```
chrsrc -c IBM.PeerNode OpQuorumTieBreaker="NetworkTieBreaker"
```

Further information:

- System Automation for Multiplatforms quorums are documented in *Tivoli System Automation for Multiplatforms: Administrator's and User's Guide*, SC34-2698.
- The commands that are related to quorums are documented in *Tivoli System Automation for Multiplatforms Installation and Configuration Guide*, SC34-2699.

Chapter 10. Verifying your implementation on z/OS

Read this information to learn about verification procedures and failover scenarios for an implementation on z/OS. Also, there is a description how to perform problem determination for SA z/OS and for each of the critical SAP components.

It contains these main topics:

- “Verification procedures and failover scenarios”
- “Problem determination methodology” on page 231

Verification procedures and failover scenarios

This topic describes the test scenarios that have been designed to test the current SA z/OS **SAPSRV add-on policy* described in this publication.

The **SAPSRV add-on policy* is based on the standard SAP infrastructure. This means that SAP's start service `sapstartsrv` and the `sapstart` process of the SAP Central Services and Replication instances are modeled and automated by SA z/OS. This topic describes test cases, which verify that SA z/OS automation and SAP have been correctly set up.

The sample screens for the described test scenarios were created in a dual-stack SAP system with SID HA2 and SAP application servers on AIX. The SID HA2 is part of the names of the sample SA z/OS resources in the following sections. The Sysplex (COHPLEX) used for testing is made up of three z/OS LPARs, COH1 (running on CEC1), COH2 and COH3 (running on CEC2).

The SA z/OS policies have been generated with the SAP High Availability Wizard for IBM Tivoli System Automation for z/OS.

Overview of the test scenarios

Read about the design, classification, and prerequisites for running the described test scenarios to verify the **SAPSRV add-on policy*.

Before you run the test scenarios that have been designed to test and verify the SA z/OS **SAPSRV add-on policy*, make sure that:

- z/OS and the network have been configured.
- SA z/OS and NetView have been configured.
- The high availability solution including the SA policy has been installed and activated.
- The SAP system is running fully functional.

Test scenario classification

The scenarios cover both *planned outages* (or planned activities) and *unplanned outages* (or failures). For each category, the 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 19 on page 194 describes, in the form of a matrix, some test scenario examples.

Table 19. 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

The SAP High Availability (HA) Test Tool (HATT)

You can perform all test scenarios listed in “Test scenarios to verify the SA z/OS policy” manually as described in this topic. However, the SAP High Availability (HA) Test Tool (HATT) performs a selection of these scenarios automatically. The following information helps you to decide to what extent you can use this tool instead of manually performing these tests.

The SAP HA Test Tool supports individual failure conditions - planned and unplanned - under workload. The tool starts a configurable workload and injects a specific failure. The level of workload, the failure and a variety of other parameters can be configured to fit many if not all SAP environments. The tool provides logs and performance numbers to help document the complete failure test. For access to the SAP HA Test Tool and for details about the tool refer to the following information:

- SAP Community: *Test Tool for High Availability Environments*
- *SAP Note 2081226: Regulations to get the HA Test Tool*

For a hands-on assistance on how to test and verify the high availability of your SAP solution with the SAP HA Test Tool refer to the publication *SAP on IBM z Systems HA Verification with SAP HA Test Tool* available on the SAP Community Network (SCN):

<http://go.sap.com/documents/2016/06/ba884016-767c-0010-82c7-eda71af511fa.html>

The mentioned publication covers a predefined set of HA use cases. The sample use cases can verify various aspects of high availability of a SAP on IBM z Systems business continuity environment like the one described in this document.

SAP on IBM z Systems HA Verification with SAP HA Test Tool demonstrates the following samples or scenarios:

- an SAP SCS instance failure event
- an SAP rolling kernel switch (RKS) event under workload
- a DB2 failure event
- ASCS maintenance

Test scenarios to verify the SA z/OS policy

To verify the SA z/OS policy, a list of test scenarios has been created, which includes planned and unplanned outages.

Planned outage scenarios

- Controlled operator intervention against SAP-related components:
 - Start and stop of all the SAP-related components
 - Start and stop of Java SCS Group (comprising the SAP start service, the start process sapstart, the central services VIPA, the enqueue and message servers, and the SAP gateway)
 - Start and stop of ABAP SCS Group (comprising the SAP start service, the start process sapstart, the central services VIPA, and the enqueue and message servers)
 - Start and stop of all resources which belong to one SAP System (enqueue replication server, ABAP SCS Group or Java SCS Group, DB2 subsystems)
 - Move of the ABAP SCS Group, or Java SCS Group, or both
 - Start and stop of the ABAP, or Java enqueue replication server, or both
 - Move of the ABAP, or Java enqueue replication server, or both from one LPAR to another (if more than two LPARs)
 - Start and stop of SAP host agent
 - Start and stop of the SAP Web Dispatcher
 - Start and stop of the SAP Solution Manager Diagnostics Agent (SMDA)
 - 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 SAP application server on Linux on z Systems
 - Start and stop of an SAP application server remotely managed from SA z/OS.
- 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
 - Shutdown and restart of the LPAR, which contains the Primary Automation Manager (PAM), where NFS server is running
 - Shutdown and restart of the LPAR, which contains the Secondary Automation Manager (SAM), where enqueue replication server is running

Unplanned outage scenarios

- Failure of an SAP component:
 - The enqueue server for ABAP, or Java, or both
 - The enqueue replication server for ABAP, or Java, or both
 - The message server for ABAP, or Java, or both
 - The SAP start process sapstart for ABAP, or Java, or both
 - The SAP start service sapstartsrv for ABAP, or Java, or both
 - An SAP application server on Linux on z Systems
 - An SAP application server, managed remotely from SA z/OS
 - A DB2 subsystem
 - The NFS server
 - The SAP gateway

- saprouter
- The SAP host agent process
- The SAP Web Dispatcher
- The SAP Solution Manager Diagnostics Agent (SMDA)
- Failure of a network component:
 - A TCP/IP stack on z/OS
 - OSPF (OMPROUTE)
 - A network adapter on z Systems
 - 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
 - the LPAR, which contains the Primary Automation Manager (PAM), where enqueue server and NFS server are running
 - the LPAR, which contains the Secondary Automation Manager (SAM)

Sample test scenarios

The following test scenarios represent a subset of the complete list that was described in the previous section. This subset focuses on the most important ones and these tests are described in the following sections. The description shows sample commands, screen displays and listings for SAP system HA2.

Planned outage scenarios

- Controlled operator intervention against SAP-related components:
 - Start and stop all components of one SAP system on z/OS
- 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, the NFS server and the PAM are running

Unplanned outage scenarios

- Failure of a critical SAP component:
 - The SAP start service sapstartsrv
 - The SAP start process sapstart
 - The ABAP, or Java enqueue server, or both
 - The ABAP, or Java message server, or both
 - The ABAP or Java enqueue replication 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 servers and NFS server are running

Test methodology

Although each scenario is different, many of the steps that need to be executed before, during, and after a test, are similar. This topic describes these steps in the form of a methodology, which you can apply to any scenario you want to test in your own environment.

Purpose of a test

The purpose of a test is characterized by the following 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 you want to test a normal startup or shutdown, a controlled movement, or do you want to simulate a failure?

Expected behavior

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 SAP application servers?
- What about transparency for the running SAP workload?

Setting up the test environment

You must already know which resources must be stopped, which must be running, and in which LPAR each component must be running when you prepare the test environment.

Verifying the resource status

Before each test, use the following checklist to review the status of all the SAP-related resources defined in SA z/OS:

1. Do all the resources monitored by SA z/OS have a compound status SATISFACTORY?

Tip: The SA z/OS command INGLIST SAP_X/APG displays the status of the application group SAP_X . If the compound status is SATISFACTORY, then you know that all resources belonging to that group have a compound state SATISFACTORY. Otherwise, you can drill down the tree of resources using option G (Members).

The following is a sample output of the SA z/OS command INGLIST SAP_X/APG, showing the application group SAP_X with a compound status of SATISFACTORY:

```
INGKYST0          SA z/OS - Command Dialogs          Line 1    of 50
Domain ID   = IPXFO          ----- INGLIST -----          Date = 10/23/15
Operator ID = HEIKES          Sysplex = COHPLEX          Time = 16:08:32
A Update   B Start   C Stop   D INGRELS  E INGVOTE  F INGINFO  G Members
H DISPTRG  I INGSCHED J INGGROUP K INGCICS  L INGIMS   M DISPMTR  T INGTWS
U User     X INGLKUP  / scroll
CMD Name   Type System  Compound   Desired   Observed   Nature
-----
SAP_X     APG          SATISFACTORY  AVAILABLE  AVAILABLE  BASIC
```

2. Are there any outstanding votes in SA z/OS?

Tip: The SA z/OS command INGVOTE displays the list of all the votes in the system. As prerequisite for processing the verification tests, the list should be empty.

3. Are there any outstanding excludes in SA z/OS?

There is no command to display all the excludes in SA z/OS at the same time. Instead of typing multiple INGINFO commands, it is recommended to use a special-purpose REXX procedure called SANCHK to display and remove all the outstanding excludes in SA z/OS. The source code for this REXX procedure can be found as file sanchkv1.txt in the file zSAP_BusinessContinuity.zip (see Chapter 14, "Reference of the z/OS high availability scripts," on page 285). You can execute this procedure directly on the command line within NetView if you add it as a member to a data set that is listed in the NetView DSICLD data definition concatenation. Check the NetView startup procedure's JCL DD statement for the DSICLD and add it to a data set (for example, in this environment, the data set USER.CNMCLST).

The following shows the output of the REXX procedure SANCHK. It shows that group SAPHA2AER_X is excluded on COH3.

```
* IPXFO    SANCHK
| IPXFO    Gathering data step 1 ...
| IPXFO    Gathering data step 2 ...
| IPXFO
|-----
Group      = SAPHA2AER_X/APG
  Excluded = COH3
  Avoided  =
|-----
End of Sanity Check
|-----
```

You can use the REXX procedure SANCHK with the option CLEAR to remove all the excludes:

```
* IPXFO    SANCHK CLEAR
| IPXFO    Gathering data step 1 ...
| IPXFO    Gathering data step 2 ...
| IPXFO    Processing CLEAR ...
| IPXFO    Processing CLEAR for SAPHA2AER_X/APG
U IPXFO    AOF099I FUNCTION COMPLETED
| IPXFO    Finished CLEAR processing
|-----
```

If you do not want to use the SANCHK procedure, you must issue individual INGINFO commands against every application group defined as SYSPLEX/MOVE groups.

In this configuration, the following commands are used:

```
INGINFO NFS_SERV_X
INGINFO SAPHA2AER_X
INGINFO SAPHA2ASC5X
INGINFO SAPHA2JER_X
INGINFO SAPHA2JSC5X
INGINFO SAPSYSRTE_X
```

The following shows a sample output of the SA z/OS command INGINFO. Look specifically at the section Group Details near the end of the output. It shows that COH3 is in the exclude list of the application group SAPHA2AER_X.


```

INGKYIN0          SA z/OS - Command Dialogs      Line 33  of 3287
Domain ID  = IPXFO  ----- INGINFO -----      Date = 10/23/15
Operator ID = HEIKES          Sysplex = COHPLEX          Time = 16:04:16

Resource => SAPHA2AER_X/APG          format: name/type/system
Target   =>          System name, domain ID or sysplex name

...
...

Group Details...
Nature           : MOVE
Move Mode        : SERIAL
Prepare Move     : YES
Failed           : Yes
Members...
...
...

Policy           :
PASSIVE = NO
EXCLUDE = COH3

```

Generally, you do not want any excludes before the test. Therefore, this exclude should be removed by issuing the SA z/OS command INGGROUP, as shown:

```
INGGROUP SAPHA2AER_X/APG ACTION=INCLUDE SYSTEMS=COH3
```

- Where are the SAP Central Services, the SAP enqueue replication server, the SAP start services, and the NFS server running before the test?

The following is a sample screen from SA z/OS showing the SAP central services and enqueue replication services for the SAP System HA2. Based on the naming conventions used by the SAP HA wizard, the command for listing the SAP Central Services resource groups is INGLIST S*HA2*/APG.

Sample command output of INGLIST S*HA2*/APG:

```

INGKYST0          SA z/OS - Command Dialogs      Line 1    of 21
Domain ID  = IPXFO  ----- INGLIST -----      Date = 10/23/15
Operator ID = HEIKES          Sysplex = COHPLEX          Time = 13:34:13
A Update   B Start   C Stop    D INGRELS  E INGVOTE  F INGINFO  G Members
H DISPTRG  I INGSCHED J INGGROUP K INGCICS  L INGIMS   M DISPMTR  T INGTWS
U User     X INGLKUP  / scroll

CMD Name      Type System  Compound  Desired  Observed  Nature
-----
SAPHA2_X      APG          SATISFACTORY AVAILABLE AVAILABLE BASIC
SAPHA2ACV_X  APG          SATISFACTORY AVAILABLE AVAILABLE MOVE
SAPHA2AENQX  APG          SATISFACTORY AVAILABLE AVAILABLE BASIC
SAPHA2AER_X  APG          SATISFACTORY AVAILABLE AVAILABLE MOVE
SAPHA2AERS   APG COH1       SATISFACTORY UNAVAILABLE SOFTDOWN BASIC
SAPHA2AERS   APG COH2       SATISFACTORY UNAVAILABLE SOFTDOWN BASIC
SAPHA2AERS   APG COH3       SATISFACTORY AVAILABLE  AVAILABLE BASIC
SAPHA2ASCS   APG COH1       SATISFACTORY UNAVAILABLE SOFTDOWN BASIC
SAPHA2ASCS   APG COH2       SATISFACTORY AVAILABLE  AVAILABLE BASIC
SAPHA2ASCS   APG COH3       SATISFACTORY UNAVAILABLE SOFTDOWN BASIC
SAPHA2ASCSX  APG          SATISFACTORY AVAILABLE  AVAILABLE MOVE
SAPHA2JCV_X  APG          SATISFACTORY AVAILABLE  AVAILABLE MOVE
SAPHA2JENQX  APG          SATISFACTORY AVAILABLE  AVAILABLE BASIC
SAPHA2JER_X  APG          SATISFACTORY AVAILABLE  AVAILABLE MOVE
SAPHA2JERS   APG COH1       SATISFACTORY UNAVAILABLE SOFTDOWN BASIC
SAPHA2JERS   APG COH2       SATISFACTORY UNAVAILABLE SOFTDOWN BASIC
SAPHA2JERS   APG COH3       SATISFACTORY AVAILABLE  AVAILABLE BASIC
SAPHA2JSCS   APG COH1       SATISFACTORY UNAVAILABLE SOFTDOWN BASIC
SAPHA2JSCS   APG COH2       SATISFACTORY AVAILABLE  AVAILABLE BASIC
SAPHA2JSCS   APG COH3       SATISFACTORY UNAVAILABLE SOFTDOWN BASIC
SAPHA2JSCSX  APG          SATISFACTORY AVAILABLE  AVAILABLE MOVE
SHA2A_SRV_X  APG          SATISFACTORY AVAILABLE  AVAILABLE MOVE
SHA2ARSRV_X  APG          SATISFACTORY AVAILABLE  AVAILABLE MOVE
SHA2J_SRV_X  APG          SATISFACTORY AVAILABLE  AVAILABLE MOVE
SHA2JRSRV_X  APG          SATISFACTORY AVAILABLE  AVAILABLE MOVE

```

The SA z/OS command for the NFS server is INGLIST NFS*. See the corresponding sample command output:

```

INGKYST0          SA z/OS - Command Dialogs          Line 1    of 7
Domain ID = IPXFO          ----- INGLIST -----          Date = 10/23/15
Operator ID = HEIKES          Sysplex = COHPLEX          Time = 15:41:40
A Update  B Start  C Stop   D INGRELS  E INGVOTE  F INGINFO  G Members
H DISPTRG I INGSCHED J INGGROUP K INGCICS  L INGIMS   M DISPMTR  T INGTWS
U User    X INGLKUP / scroll
CMD Name   Type System   Compound   Desired   Observed   Nature
-----
NFS_SERV_X APG           SATISFACTORY AVAILABLE AVAILABLE MOVE
NFSCLIENT APL COH1      SATISFACTORY AVAILABLE AVAILABLE
NFSCLIENT APL COH2      SATISFACTORY AVAILABLE AVAILABLE
NFSCLIENT APL COH3      SATISFACTORY AVAILABLE AVAILABLE
NFSSERV    APL COH1      SATISFACTORY UNAVAILABLE SOFTDOWN
NFSSERV    APL COH2      SATISFACTORY AVAILABLE AVAILABLE
NFSSERV    APL COH3      SATISFACTORY UNAVAILABLE SOFTDOWN

```

5. Are the DB2 subsystems available?

Based on the naming conventions used by the SAP HA wizard, the command for listing the DB2 resources is INGLIST HA2*.

Sample command output of: INGLIST HA2*

```

INGKYST0          SA z/OS - Command Dialogs          Line 1    of 28
Domain ID = IPXFO          ----- INGLIST -----          Date = 03/29/10
Operator ID = HEIKES          Sysplex = COHPLEX          Time = 16:52:40
A Update  B Start  C Stop   D INGRELS  E INGVOTE  F INGINFO  G Members
H DISPTRG I INGSCHED J INGGROUP K INGCICS  L INGIMS   M DISPMTR  T INGTWS
U User    X INGLKUP / scroll
CMD Name   Type System   Compound   Desired   Observed   Nature
-----
HA2_X      APG           SATISFACTORY AVAILABLE AVAILABLE SERVER
HA21       APG COH1      SATISFACTORY AVAILABLE AVAILABLE BASIC
HA21_LITE_X APG           SATISFACTORY UNAVAILABLE SOFTDOWN MOVE
HA21_X     APG           SATISFACTORY AVAILABLE AVAILABLE MOVE
HA21DBM1  APL COH1      SATISFACTORY AVAILABLE AVAILABLE
HA21DIST  APL COH1      SATISFACTORY AVAILABLE AVAILABLE
HA21IRLM  APL COH1      SATISFACTORY AVAILABLE AVAILABLE
HA21LITE  APL COH2      SATISFACTORY UNAVAILABLE SOFTDOWN
HA21LITE  APL COH3      SATISFACTORY UNAVAILABLE SOFTDOWN
HA21MSTR  APL COH1      SATISFACTORY AVAILABLE AVAILABLE
HA22       APG COH2      SATISFACTORY AVAILABLE AVAILABLE BASIC
HA22_LITE_X APG           SATISFACTORY UNAVAILABLE SOFTDOWN MOVE
HA22_X     APG           SATISFACTORY AVAILABLE AVAILABLE MOVE
HA22DBM1  APL COH2      SATISFACTORY AVAILABLE AVAILABLE
HA22DIST  APL COH2      SATISFACTORY AVAILABLE AVAILABLE
HA22IRLM  APL COH2      SATISFACTORY AVAILABLE AVAILABLE
HA22LITE  APL COH1      SATISFACTORY UNAVAILABLE SOFTDOWN
HA22LITE  APL COH3      SATISFACTORY UNAVAILABLE SOFTDOWN
HA22MSTR  APL COH2      SATISFACTORY AVAILABLE AVAILABLE
HA23       APG COH3      SATISFACTORY AVAILABLE AVAILABLE BASIC
...

```

6. Are the NFS file systems accessible that are mounted on the remote SAP application server?

- **Verify that the file systems are exported:**

On the z/OS UNIX command line use the command
/usr/sbin/showmount -e <sapnfsv>

For AIX or Linux application servers use the command
showmount -e <sapnfsv>

- **Verify that the file systems are mounted:**

From an AIX or Linux application server use the command
df

- **Verify that the mounted file systems are accessible:**

To verify that you have access to the file systems from the SAP application server, use the following command on AIX or Linux :

```
ls -a1R /sapmnt/HA2/*
```

- **Verify that the mounted file systems are accessible from SAP:**

Log on to the SAP system and use SAP transaction AL11 to verify that you can access the files in the SAP directories.

Preparing for the test

During the unplanned outage scenarios, you want to verify the impact of the failure for end users and for any workload that is running on the system. Therefore, perform the following preparation steps before each test:

1. Log on to all the SAP application servers.
2. Create an SAP workload:
 - For testing planned outages, the SAP local client copy is a good scenario. The *Client Copy* function should not fail during those planned outages tests, where the AS to DB connection is not breaking, or where the AS database connection is redirected to another DB2 data sharing member prior to a data sharing member outage. Moving AS database connections to a standby DB member using SAP transactions ST04 or DBACOCKPIT, keeps SAP in a consistent transaction state.
 - For unplanned outage tests, SAP transaction SGEN is a good candidate. All SGEN processes should run on as long as the SGEN master process is not affected by the outage. SGEN is able to handle DB connection losses transparently.
3. Generate entries in the ABAP enqueue table.

Tip: Use transaction SM12 to generate entries in the ABAP enqueue table.

From the primary panel of transaction SM12, select **Lock Entries**, enter *test* in the transaction field, as shown in the following window:

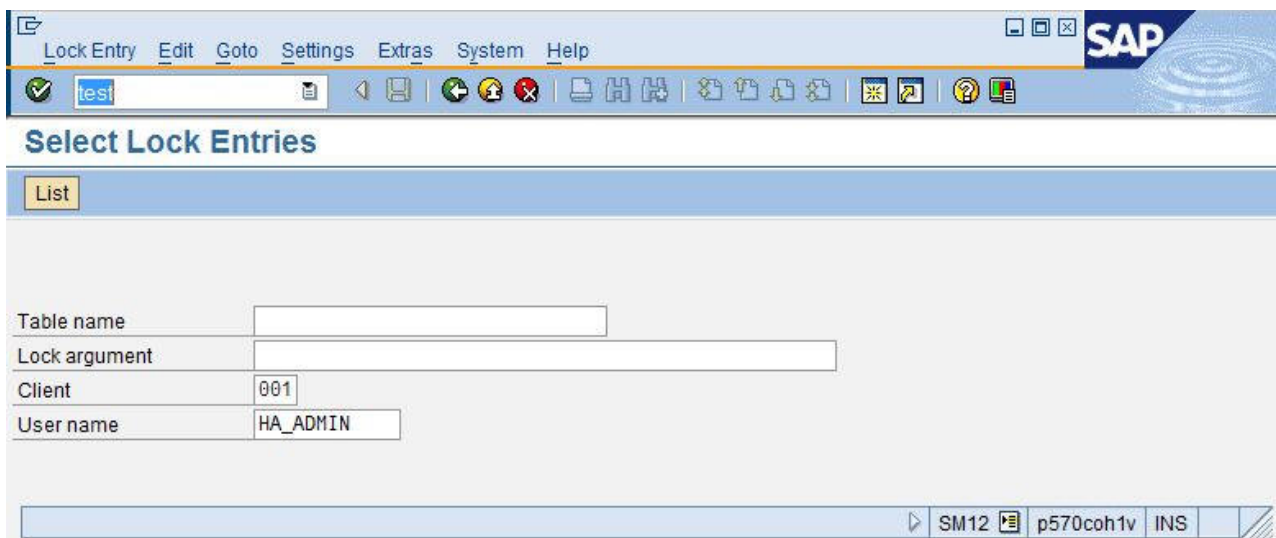


Figure 40. SM12 primary window

Press **Enter**. A new selection is displayed in the menu bar, **Error handling**.

Click **Error handling** → **Test tools** → **Mass calls**

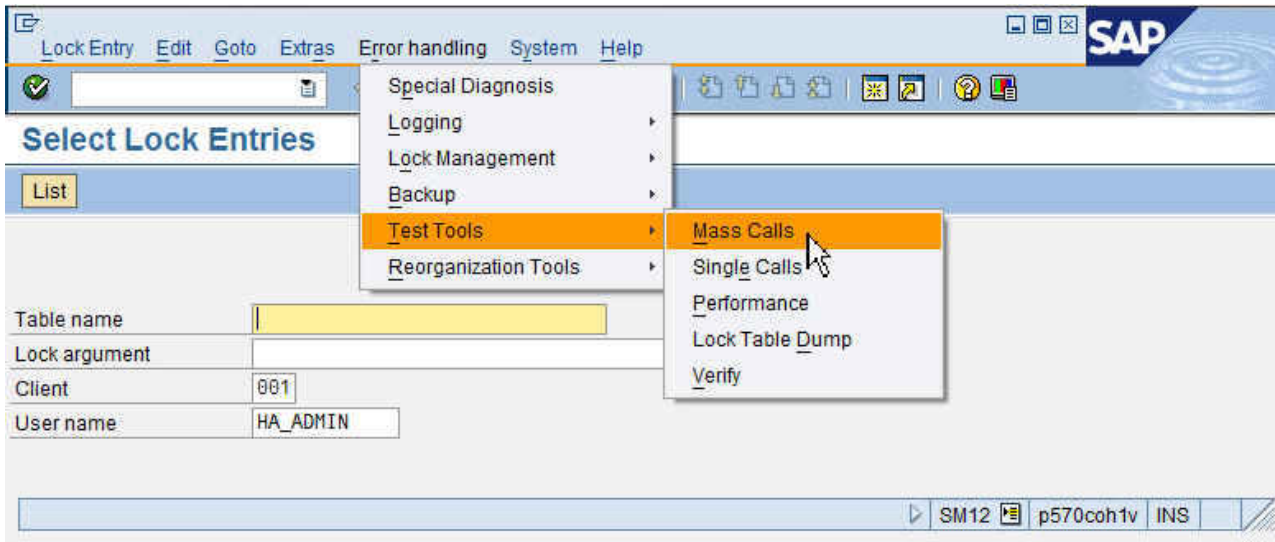


Figure 41. Error handling menu

Choose the number of lock entries you want to create (for example, you can use the default of 10 lock entries). Click **Execute**:

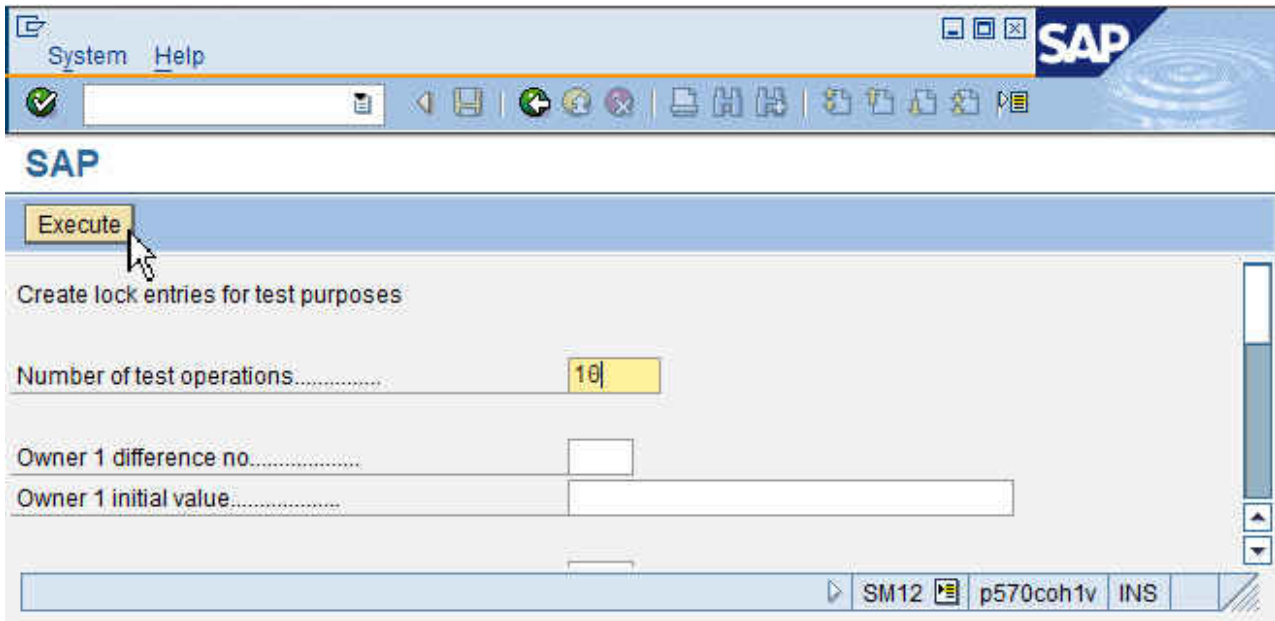


Figure 42. Enqueue test: start mass enqueue operations

The screen must stay open for the duration of the test. From another screen, use SM12 to list the entries in the enqueue table:

The screenshot shows the SAP Lock Entry List interface. The title bar includes 'Lock Entry', 'Edit', 'Goto', 'Settings', 'Extras', 'System', and 'Help'. The main window title is 'Lock Entry List'. Below the title bar is a toolbar with icons for Refresh, Details, and various data manipulation functions. The main area contains a table with the following data:

Client	User name	Time.....	Lock mod...	Table	Argument	Use Count.	Use Count.
001	HA_ADMIN	16:49:21	E	GRA 0	ARG 0	0	1
001	HA_ADMIN	16:49:21	E	GRA 1	ARG 1	0	1
001	HA_ADMIN	16:49:21	E	GRA 2	ARG 2	0	1
001	HA_ADMIN	16:49:21	E	GRA 3	ARG 3	0	1
001	HA_ADMIN	16:49:21	E	GRA 4	ARG 4	0	1
001	HA_ADMIN	16:49:21	E	GRA 5	ARG 5	0	1
001	HA_ADMIN	16:49:21	E	GRA 6	ARG 6	0	1
001	HA_ADMIN	16:49:21	E	GRA 7	ARG 7	0	1
001	HA_ADMIN	16:49:21	E	GRA 8	ARG 8	0	1
001	HA_ADMIN	16:49:21	E	GRA 9	ARG 9	0	1

Below the table, it says 'Selected Lock Entries: 10'. At the bottom right, the status bar shows 'SM12', 'p570coh1v', and 'INS'.

Figure 43. List of entries in the enqueue table

You can also verify the enqueue table entries on the operating system level of your SAP application server with SAP utility *enqt*:

```
p570coh2:ha2adm 1> enqt pf=/usr/sap/HA2/SYS/profile/HA2_ASCS20_ha2ascsv 20
```

4. Generate entries in the Java enqueue table.

Using a browser, logon to the *SAP NetWeaver Administrator* tool. Then select the tab Availability and Performance Management. On the Resource Monitoring page click on *Locks*.

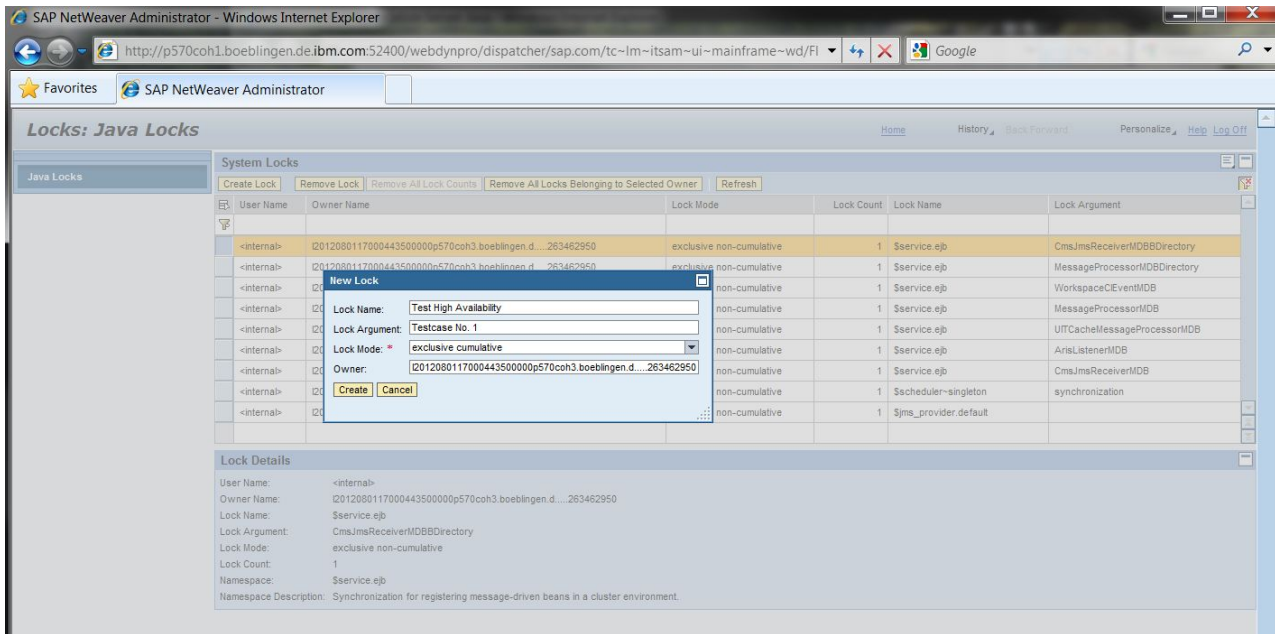


Figure 44. Create Java locks

On the System Locks page you can now click on Create Lock and create your own Java test lock(s).

Figure 45 shows the list of existing Java locks. On the same page you may delete your test your test lock(s) later.

You can also verify the enqueue table entries on the operating system level of

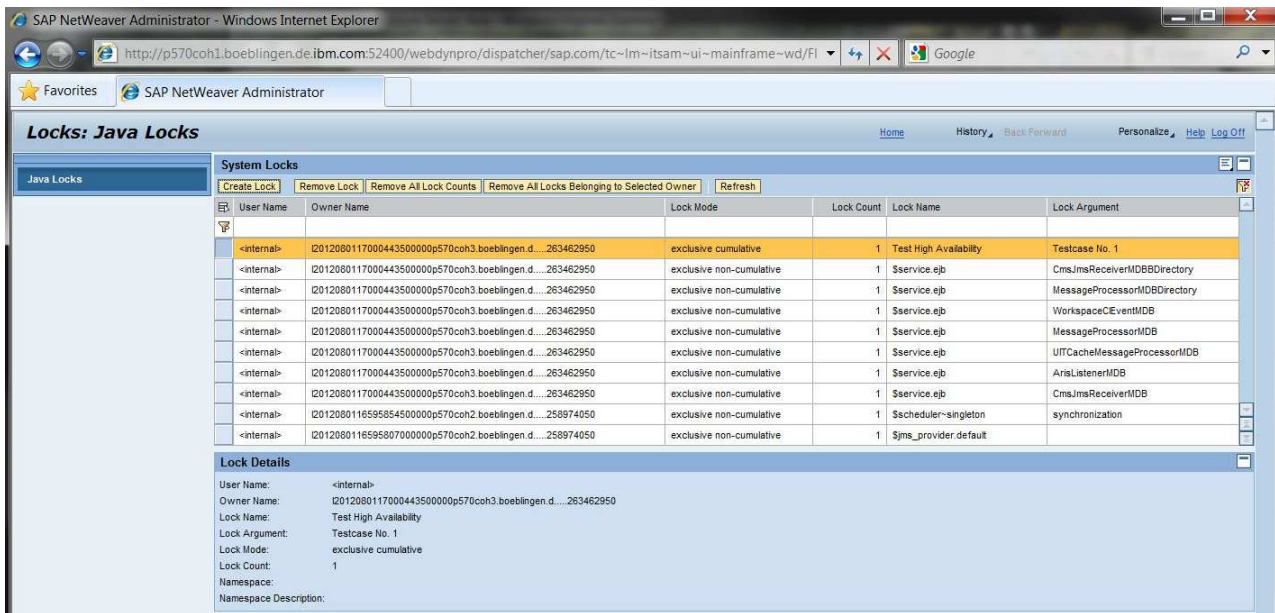


Figure 45. List of Java locks

your SAP application server with SAP utility *enqt*:

```
p570coh2:ha2adm 2> enqt pf=/usr/sap/HA2/SYS/profile/HA2_SCS21_ha2scsv 20
```

Running the test

The initiation of the test depends on the type of scenario.

- For a planned outage or a controlled move of resources, SA z/OS must be used for the following tasks:
 - Starting and stopping of resources
 - Moving of resources
 - Excluding resources on specific systems
 - Initiating SA z/OS 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

Verifications after the test

After each test, first review the status of all the components using the same checklist as the one used before the test (see “Verifying the resource status” on page 197).

Then, depending on the type of scenario (usually in the case of a failure), perform some additional verifications, such as:

- looking at the SAP system log (SAP transaction SM21)
- searching the SAP developer trace files for error messages in the work directory of the SAP instance running those services:
 - *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, or for database connection problems
- displaying the status of internal and TCP/IP connections (SAP transaction SM59)
- checking whether the workload that was created is still running (SAP transaction SM66)
- checking the number of lock entries in the enqueue table (SAP transaction SM12)
- displaying the DB2 threads using the DB2 command `-DIS THREAD(*)`

Note: A trace file called *enquelog* in the central services instance work directory logs the activity of the enqueue server and the status of the replication.

Analyzing problems

If the results differ from the expected behavior, it is necessary to understand why. The following section contains some tips to help you with this complex troubleshooting phase in “Problem determination methodology” on page 231).

Planned outage test scenarios

Read information about planned outage test scenarios which you can perform to verify the SA z/OS *SAPSRV *add-on policy*.

This topic describes the following test scenarios:

- “Automatic restart of SAP start service sapstartsrv”
- “Stopping and starting all SAP components on z/OS” on page 207
- “Stopping and starting the SAP infrastructure group” on page 209
- “Stopping and starting all SAP application servers” on page 211
- “Stopping and starting a single SAP application server” on page 212
- “Shutting down and restarting an LPAR” on page 212
- “Starting up all LPARs one after the other” on page 215

For each scenario, you can specify the following:

- Purpose of the test
- Expected behavior
- Initial setup
- Phases of the execution
- Observed results

“Verifying the resource status” on page 197 describes verification tasks that are performed before and after each test to check the status of the SAP-related components. In this section, these steps are not repeated. However, the description of each test may contain additional verification tasks that are specific to the scenario.

Automatic restart of SAP start service sapstartsrv

In this scenario, you test the restart of the SAP start service sapstartsrv. A restart of sapstartsrv occurs for example, if a new SAP kernel level is copied into the local executable directory/usr/sap/<sid>/<instance>/exe by SAP's sapcpe mechanism as part of a restart of the instance. A running sapstartsrv process will restart itself within five minutes after a new executable file is detected. The test simulates this behavior and verifies that SA z/OS handles a restart of sapstartsrv correctly. Table 20 summarizes the execution of the sapstartsrv restart.

For more information on the behavior of sapstartsrv, see the SAP documentation SAPStartSRV in HA Environments.

Table 20. Automatic restart of sapstartsrv after patch level upgrade

Scenario characteristics	Description
Purpose	Scope: All SAP HA2 components on z/OS Action: trigger a restart of sapstartsrv for the ABAP or Java central services instance. Issue the following command from the LPAR where the sapstartsrv process is currently running. Note the process ID before and after the restart. <code>sapcontrol -nr <instno> -function RestartService</code>

Table 20. Automatic restart of sapstartsrv after patch level upgrade (continued)

Scenario characteristics	Description
Expected behavior	The sapstartsrv process restarts itself under a new process ID. SA z/OS detects the successful restart and continues to show the sapstartsrv resource SAPHA2A_SRV (for ABAP) or SAPHA2J_SRV (for Java) as AVAILABLE after the restart.
Setup	COH1, COH2 and COH3 must be running, including all required z/OS resources and SAP-related resources.
Observed results, if unexpected:	

Stopping and starting all SAP components on z/OS

In this scenario you test the stop and restart of all SAP HA2 components on z/OS (including ABAP and Java SAP central services instances, ABAP and Java enqueue replication server instances, database servers, and so on).

The following table summarizes the execution of the stop phase.

Table 21. Stop of all SAP HA2 components with SA z/OS

Scenario characteristics	Description
Purpose	Scope: All SAP HA2 components on z/OS Action: Planned stop using SA z/OS
Expected behavior	All HA2-related resources should stop correctly. The NFS server and SAProuter continue to run.
Setup	COH1, COH2 and COH3 must be running, including all required z/OS resources and SAP-related resources.
Execution	Stop the SAP application servers before stopping the SAP components on z/OS UNIX. Then issue a STOP request in SA z/OS against the application group SAPHA2_X.
Observed results, if unexpected:	

Table 22 summarizes the execution of the start phase.

Table 22. Start of all SAP HA2 components with SA z/OS

Scenario characteristics	Description
Purpose	Scope: All SAP HA2 components on z/OS Action: Planned start using SA z/OS
Expected behavior	All HA2-related resources start up correctly.
Setup	COH1, COH2 and COH3 must be running, with all required z/OS resources, but all HA2-related resources are stopped.
Execution	Kill the STOP request in SA z/OS against the application group SAPHA2_X.

Table 22. Start of all SAP HA2 components with SA z/OS (continued)

Scenario characteristics	Description
Observed results, if unexpected:	

To stop all SAP HA2 components on z/OS, issue a STOP request against the application group SAPHA2_X (option C):

```

INGKYSTO                SA z/OS - Command Dialogs      Line 1    of 81
Domain ID = IPXFO      ----- INGLIST -----      Date = 09/18/12
Operator ID = HEIKES   Sysplex = COHPLEX             Time = 15:15:06
  A Update  B Start   C Stop   D INGRELS  E INGVOTE  F INGINFO  G Members
  H DISPTRG I INGSCHED J INGGROUP K INGCICS L INGIMS  M DISPMTR T INGTSW
  U User    X INGLKUP / scroll
CMD Name      Type System  Compound      Desired      Observed      Nature
-----
c  SAPHA2_X   APG          SATISFACTORY AVAILABLE     AVAILABLE     BASIC
   SAPHA2A_SRV APL COH1     SATISFACTORY UNAVAILABLE  SOFTDOWN
   SAPHA2A_SRV APL COH2     SATISFACTORY AVAILABLE     AVAILABLE
   SAPHA2A_SRV APL COH3     SATISFACTORY UNAVAILABLE  SOFTDOWN
  
```

In this example, you want a normal stop of the SAP HA2 system on z/OS. Thus, stay with the default type NORM.

Note: If you do not stop the SAP HA2 application servers prior to stopping SAPHA2_X, then any running SAP ABAP transactions break, SAPGUI users are disconnected and their transactions disrupted. The SAP application servers stay idle until the SAP resources on z/OS are running again. As soon as SAP resources on z/OS are up again, ABAP processes reconnect and SAPGUI login is possible again. Java application servers can successfully reconnect only if the Java resources on z/OS are back before the reconnect time-out for the Java message server. If you want to stop the remote SAP application server, you must either issue a stop request in your automation software (System Automation for Multiplatforms), or, if you did not set up automation control for your SAP application servers, you must manually stop the SAP application server instances.

If your SAP application servers are managed remotely by SA z/OS, and if you have changed or set up the remote application server policy to use the REXX script SAPRASTP in the SHUTNORM policy definition, then a *STOP vote* on group SAPHA2_X with TYPE=NORM (default) is sufficient to stop the remote application servers and the rest of the SAP system.

If you do not use SAPRASTP, then you must stop the SAP system in two steps:

1. First issue the STOP request with option FORCE to stop the corresponding SAPHA2RAS_X group. A stop with TYPE=NORM (default) without SAPRASTP would only stop the z/OS monitoring of the remote SAP application server. The SAP application server itself would continue to run.
2. Then continue to stop SAPHA2_X to stop the rest of the SAP system.

The status of the HA2 related resources changes from AVAILABLE via STOPPING to AUTODOWN. The NFS server and SAProuter are still running.

To restart the SAP system, you must kill the remaining *MakeUnavailable* vote on the application group SAPHA2_X:

```

INGKYRQ0          SA z/OS - Command Dialogs          Line 1 of 10
Domain ID = IPXFP ----- INGVOTE -----          Date = 02/05/10
Operator ID = HEIKES          Sysplex = COHPLEX          Time = 13:40:39

Resource ==> SAPHA2_X/APG/COH1
System ==>          System name, domain id or sysplex name

Desired Available...: Always

Cmd: C cancel request  K Kill request  S show request details
Cmd Action WIN Request/Vote Data
-----
STOP  Y Request   : MakeUnAvailable
      Created    : 2010-02-05 13:16:03
      Originator  : OPERATOR(HEIKES)
      Priority    : 01720000      Should Be Down - Operator
      Status     : Winning/Satisfied

```

After some time, all SAP HA2-related resources are running again. The resources are in AVAILABLE status and the SAP enqueue server and SAP enqueue replication server are running on different LPARs (for example, SAP central services on COH1 and the SAP enqueue replication server on COH2).

Now you can restart the SAP application servers if required. Use automation software (System Automation for Multiplatforms), or, if you did not set up automation control for your SAP application servers, you must manually start the SAP application server instances. If your SAP application servers are managed remotely by SA z/OS then cancel the stop vote on the SAPHA2RAS_X group in SA z/OS.

Stopping and starting the SAP infrastructure group

When you run this test scenario, you must have in mind the relationships between members of the SAP infrastructure group SAPSYS_X and the SAP application servers that are managed as proxy resources from SA z/OS. The NFS server must not be stopped until all SAP application servers of each or all SAP systems have been stopped. This condition is implemented in the *HasParent* relationship between the NFS server group NFS_SERV_X and the application server proxy resources group SAP<SID>RAS_X. As a prerequisite for this test, ensure that you have implemented the new STOP mechanism via the SAPRASTP script in your application server proxy resources (see “Policy configuration for ABAP-only and Java-only application servers” on page 176). This test scenario models, for example, the maintenance of the NFS server if you must have the NFS server down on all LPARs at the same time.

Caution: All SAP application servers are stopped as part of this test, so all SAP systems become unavailable during that time.

Table 23 on page 210 summarizes the execution of the stop phase for the SAP infrastructure group.

Table 23. Stop the SAP infrastructure group

Scenario characteristics	Description
Purpose	Scope: SAP infrastructure group SAPSYS_X with all its members, in particular the NFS server group NFS_SERV_X and SAP application server proxy resources. Action: Planned stop of the SAP infrastructure group. Checks the relationships during a stop of the SAP infrastructure group.
Expected behavior	<ol style="list-style-type: none"> 1. First all SAP application servers are stopped. 2. Then the components of the SAP infrastructure group (NFS Server, SAP Router, SAP Web Dispatcher, ...) are stopped.
Setup	All components of the SAP infrastructure group and all SAP application servers must be running.
Execution	Issue a STOP request for the SAP infrastructure group SAPSYS_X.
Observed results, if unexpected:	

Verify the expected results:

- Check that the SAP infrastructure group SAPSYS_X, its members, and all application servers proxy resources are in observed status SOFTDOWN.
- Optionally, verify on operating system level that all SAP application server processes have stopped.
- Check that the DB2 resources and the SAP central services resources are not impacted and stay in status AVAILABLE.

Table 24 summarizes the execution of the restart phase of the SAP infrastructure group.

Table 24. Restart the SAP infrastructure group

Scenario characteristics	Description
Purpose	Scope: SAP application servers and SAP infrastructure group SAPSYS_X with all its members, in particular the NFS server group NFS_SERV_X Action: Planned restart of the SAP infrastructure group. Check the relationships during a (re)start of the SAP infrastructure group.
Expected behavior	<ul style="list-style-type: none"> • The SAP infrastructure group and its components start. • As soon as the NFS server has started successfully, all SAP application servers are started.
Setup	All components of the SAP infrastructure group and all SAP application servers are stopped via a STOP request for SAPSYS_X.
Execution	Trigger a restart by canceling the previous STOP request for the SAP infrastructure group SAPSYS_X.
Observed results, if unexpected:	

Verify the expected results:

- Check that the SAP infrastructure group SAPSYS_X and the SAP application servers proxy resources are in observed status AVAILABLE.

- Optionally, verify on operating system level that all SAP application server processes are started.
- Check that an SAP logon to the SAP systems is possible.

Stopping and starting all SAP application servers

This is a scenario, where, for example, system maintenance requires downtime of the application server hardware or operating system. In this scenario you test stopping and restarting of SAP HA2 application servers that are managed by the Remote AppServer support of the **SAPSRV add-on policy*. In this case the AppServers are not managed locally on their application server host by SA MP. They are managed remotely through SA z/OS. For more information on remotely managed application servers see Chapter 8, “Customizing Tivoli System Automation for z/OS,” on page 145.

Table 25 summarizes the process of the stop phase.

Table 25. Stop of all SAP HA2 application servers with SA z/OS

Scenario characteristics	Description
Purpose	Scope: All SAP HA2 application servers Action: Planned stop using SA z/OS
Expected behavior	All HA2 application servers should stop correctly.
Setup	COH1, COH2 and COH3 must be running, including all required z/OS resources and SAP-related resources. SAPHA2R* resources must be configured and active.
Execution	Issue a STOP request with TYPE=NORM in SA z/OS against the application group SAPHA2RAS_X, if you have changed or set up the remote application server policy to use the REXX script SAPRASTP in SHUTNORM. Otherwise, if you do not use SAPRASTP, then you must issue the STOP request with TYPE=FORCE in SA z/OS against the application group SAPHA2RAS_X.
Observed results, if unexpected:	

Table 26 summarizes the processing of the start phase.

Table 26. Start of all SAP HA2 components with SA z/OS

Scenario characteristics	Description
Purpose	Scope: All SAP HA2 application servers Action: Planned starting using SA z/OS
Expected behavior	All HA2 application servers start up correctly.
Setup	COH1, COH2 and COH3 must be running with all required z/OS resources, but all HA2 application servers are stopped. SAPHA2R* resources must be configured and active.
Execution	Kill the STOP request in SA z/OS against the application group SAPHA2RAS_X.

Table 26. Start of all SAP HA2 components with SA z/OS (continued)

Scenario characteristics	Description
Observed results, if unexpected:	

If you have changed or set up the remote application server policy to use the REXX script SAPRASTP in SHUTNORM, then a STOP request on group SAPHA2RAS_X with TYPE=NORM (default) is sufficient to stop all remote application servers (option C).

If you do not use SAPRASTP, then you must issue the STOP request with option FORCE to stop the corresponding SAPHA2RAS_X group. A stop with TYPE=NORM (default) without SAPRASTP would only stop the z/OS monitoring of the remote SAP application server. The SAP application server itself would continue to run.

```

INGKYST0                SA z/OS - Command Dialogs      Line 1    of 1
Domain ID = IPXFO        ----- INGLIST -----      Date = 08/09/13
Operator ID = RHIMM      Sysplex = COHPLEX           Time = 11:31:37
A Update  B Start  C Stop  D INGRELS  E INGVOTE  F INGINFO  G Members
H DISPTRG I INGSCHED J INGGROUP K INGCICS L INGIMS  M DISPMTR T INGTWS
U User    X INGLKUP / scroll
CMD Name      Type System  Compound      Desired      Observed      Nature
-----
c  SAPHA2RAS_X APG                SATISFACTORY AVAILABLE  AVAILABLE  SERVER
    
```

The status of the HA2 application server resources changes from AVAILABLE via STOPPING to SOFTDOWN.

To restart the SAP system, cancel the STOP request for the sysplex group SAPHA2RAS_X.

After some time, all SAP HA2 application servers are running again. The status of the proxy resources in SA z/OS is AVAILABLE.

Stopping and starting a single SAP application server

Not stopping and starting the whole group of SAP application servers but a single SAP application server is similar to the previous description. In this case you would drill down from the high level group SAPHA2RAS_X to the application group or application you need to stop/start. From there you can basically follow the previous description with regards to one SAP application server.

Shutting down and restarting an LPAR

In this scenario you test the shutdown and restart of the LPAR where the enqueue server and the NFS server are running. This scenario is split into two parts:

1. The shutdown of the LPAR
2. The restart of the LPAR

Table 27 on page 213 summarizes the execution of the shutdown phase.

Table 27. Shutdown of the LPAR where the ES and NFS servers are running

Scenario characteristics	Description
Purpose	Scope: One LPAR Planned shutdown of the LPAR where SA PAM, the SAP central services, and the NFS server are running.
Expected behavior	The SA PAM moves to another LPAR. The NFS server moves to the other LPAR. The SAP central services move to another LPAR. The SAP enqueue replication server stops or moves to another LPAR if more than two LPARs are available. The SAP application servers reconnect to the message server and enqueue server. The LPAR stops correctly to the point where you can enter the following command to remove the LPAR from the sysplex: <i>/V XCF,<sysname>,OFFLINE</i>
Setup	The three LPARs COH1, COH2 and COH3 must be running, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> • the SAP central services running on COH1 • the NFS server running on COH1 • the SAP enqueue replication server is running on another LPAR (COH2 or COH3)
Execution	In SAP transaction DBACOCKPIT switch the DB connection for ABAP to target DB member on COH2. Move Java database connections by stopping DDF of DB2 member on COH1 (see "DB2 connection failover" on page 15). When DB2 threads moved from DB2 member of COH1 to DB2 member on COH2, use SA z/OS and issue a STOP request against the system group COH1.
Observed results, if unexpected:	

Table 28 summarizes the execution of the restart phase.

Table 28. Restart of the LPAR where the ES and NFS servers were running previously

Scenario characteristics	Description
Purpose	Scope: One LPAR Restart after planned shutdown of the LPAR where SA z/OS Primary Automation Manager (PAM), the SAP central services and the NFS server are running (in our case COH1)

Table 28. Restart of the LPAR where the ES and NFS servers were running previously (continued)

Scenario characteristics	Description
Expected behavior	<p>COH1 starts up with all required address spaces including database server.</p> <p>The SAP central services, the SA z/OS PAM and the NFS server stay on the LPAR to where they moved during the shutdown of COH1.</p> <p>The SAP enqueue replication server stays on the LPAR to where it was moved during shutdown of COH1. Or, if only two LPARs are available, it restarts on the IPLed LPAR.</p>
Setup	<p>All SAP Central Services resources, SAP utilities and NFS server must be up on either COH2 or COH3 LPAR. Because there are still two active LPARs, ERSs run on different LPARs than the SCSs.</p> <p>COH1 must be down and HMC access is required to execute the test.</p>
Execution	IPL COH1
Verifications	If the enqueue replication server is restarted it reconnects to the enqueue server.
Observed results, if unexpected:	

All the SAP-related resources are in AVAILABLE status. The NFS server and the SAP central services are running on COH2. The SAP enqueue replication server is running on COH3

Move away those SAP application servers that are connected to the DB2 member on COH1. See “Switching database connections for a single SAP application server” on page 251 or “Switching database connections for multiple SAP ABAP application servers” on page 261 for details. Then stop the LPAR through System Automation by issuing a STOP request against the system group COH1.

COH1 shuts down to the point where you can enter the following MVS command to remove COH1 from the sysplex:

```
/V XCF,COH1,OFFLINE
```

Note: Starting with z/OS 1.12 and an active SFM policy as described in “Recommendations for sysplex failure management (SFM) policy definitions” on page 109, partitioning the LPAR COH1 is done automatically, without operator intervention.

The second part of the test can now be performed: the restarting of the LPAR COH1 from the HMC.

During the IPL process, SA z/OS starts, and according to the active SA z/OS policy, it restarts the base z/OS components, the DB2 subsystem and potentially certain SAP resources.

For more information refer to “Verifying the resource status” on page 197.

Because you 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 COH2.

Look at the enqueue server log file:

/usr/sap/HA2/ASCS20/work

to verify that the enqueue replication server reconnected to the enqueue server and that the replication was active. The following is the extract of this file corresponding to the time interval of the test.

```
RepAct: Tue Feb  2 14:29:42 2010: replication activated
      Stop: Wed Feb  3 17:00:58 2010: enqueue server stopped: normal shutdown
      Start: Wed Feb  3 19:37:22 2010: enqueue server started
RepAct: Wed Feb  3 19:39:24 2010: replication activated
```

SAP ABAP or Java instances automatically reconnect to the DB2 member on the restarted LPAR if the **affinityFallbackInterval** parameter in the db2dsdriver.cfg or config.xml configuration file is set. If you have set this parameter to a positive value, then the fallback to the primary member happens automatically after the specified number of seconds.

If the **affinityFallbackInterval** parameter is set to 0, and you have defined specific dynamic DDF aliases for your ABAP or Java instances, you can alternatively stop the dynamic DB2 DDF alias on the secondary DB2 member to trigger a reconnect of an ABAP or Java instance to its primary DB2 member.

Starting up all LPARs one after the other

In this scenario, you test the normal startup of the LPARs, one after the other.

This scenario is split into two parts:

1. The startup of the first LPAR (in this case COH1)
2. The startup of the second and third LPAR (in this case COH2 and COH3)

Table 29 summarizes the startup of the first LPAR.

Table 29. Startup of the first LPAR

Scenario characteristics	Description
Purpose	Scope: One LPAR Action: Planned startup of an LPAR while the other ones are down.
Expected behavior	The LPAR starts with all required address spaces including all SAP-related resources: database server, SAP host agent, NFS server, SAP Central Services (including VIPA for SCS, enqueue server, message server and SAP gateway) and sapstartsrv service for the SAP components on z/OS UNIX, but not SAP enqueue replication server.
Setup	All LPARs must be down. HMC access is required to execute the test.
Execution	IPL COH1
Observed results, if unexpected:	

Table 30 on page 216 summarizes the startup of the remaining LPARs.

Table 30. Startup of the remaining LPARs

Scenario characteristics	Description
Purpose	Scope: Remaining two LPARs Action: Planned startup of remaining two LPARs while one other is running
Expected behavior	The LPARs start with all required address spaces including all SAP-related resources: database server and enqueue replication server and its sapstartsrv service, and the SAP host agent.
Setup	The first LPAR must be up with all required z/OS resources and SAP-related resources: database server, SAP host agent, plus NFS server and enqueue server. The other LPARs must be down. HMC access is required to execute the test.
Execution	IPL at first COH2, then COH3
Observed results, if unexpected:	

Unplanned outage test scenarios

This topic describes the unplanned outage test scenarios you can perform to verify the **SAPSRV add-on policy*.

The failure of SAP resources and how SA z/OS reacts, depends on the severity of the failure. For example, if the enqueue or message server fails (both are a single point of failure), then the SAP system is no longer operable. This is not true for example, if the enqueue replication server fails, which has no direct impact for a running SAP system.

In order to simulate an unplanned outage of an SAP resource, two ways are used in the following:

1. Sending a kill -2 signal to the process. This simulates, for example, an operator intervention. For a z/OS UNIX process it means a normal stop for the process, allowing the process to perform cleanup actions, if any are implemented.
2. Sending a kill -9 signal to the process. This simulates, for example, a program crash. For z/OS UNIX this means that the operating system does not give back any control to the process.

This topic describes the following test scenarios:

- “Failure of the SAP enqueue server” on page 217
- “Failure of the message server” on page 220
- “Failure of the enqueue replication server” on page 221
- “Failure of the SAP start service” on page 222
- “Failure of the sapstart process of the SAP Central Services” on page 223
- “Failure of the sapstart process of the enqueue replication server” on page 225
- “Failure of the NFS server” on page 226
- “Failure of a TCP/IP stack” on page 227

- “Failure of an LPAR” on page 229

For each test scenario, the following is documented:

- Purpose of the test
- Expected behavior
- Initial setup
- Preparation for the test
- Phases of the execution
- Observed results

“Verifying the resource status” on page 197 describes the verification tasks that can be performed before and after each test to check the status of the SAP-related components. These steps are not repeated in this section. However, the description of each test may contain additional verification tasks that are specific to the scenario.

Failure of the SAP enqueue server

In this scenario you can simulate the failure of the enqueue server and test the behavior of SA z/OS. You can also measure the impact of the failure on the SAP workload.

The following table summarizes the execution of the test.

Table 31. Failure of the SAP enqueue server

Scenario characteristics	Description
Purpose	Scope: Enqueue server Action: Unplanned outage
Expected behavior	Example ABAP enqueue server: SA z/OS shows a PROBLEM or HARDDOWN status for the failed resource SAPHA2AEN and restarts its ASCS group (SAPHA2ASCS) on the LPAR where its enqueue replication server runs. Before, its corresponding SAP start service (SAPHA2A_SRV) moves, because of the <i>HasParent</i> relationship to SAPHA2ASCS. The enqueue replication server stops and restarts on COH1. Before, its corresponding SAP start service (SAPHA2AR_SR) moves to COH1 because of the <i>HasParent</i> relationship to SAPHA2AERS. The failure has no impact on the SAP workload.
Setup	COH1, COH2, and COH3 must be running, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> • the enqueue server running on COH2 • the enqueue replication server running on another LPAR (COH3)
Preparation	<ul style="list-style-type: none"> • Log on to all SAP application servers. • Create a workload on one SAP application server (on zLinux ihlscoh1). • Create entries in the enqueue table.
Execution	Use the UNIX command <code>kill</code> once with signal <code>-9</code> and once with signal <code>-2</code> to kill the enqueue server process outside of SA z/OS.

Table 31. Failure of the SAP enqueue server (continued)

Scenario characteristics	Description
Verifications	<ul style="list-style-type: none"> • 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 file <i>dev_enqserv</i>, in the developer traces <i>dev_disp</i> and <i>dev_wx</i>, and in the system log (SM21).
Observed results, if unexpected:	

Before the test, all SAP-related resources are in AVAILABLE status. The SAP enqueue server is running on COH2, and the enqueue replication server is running on a different LPAR (in this test it is in LPAR COH3).

To simulate the failure, kill the enqueue server process (en.sapHA2_ASCS20), using the UNIX command `kill -9 <pid>`.

The MVS log shows the failure of the enqueue server on COH2 and its restart on COH3:

```

COH2 12264 15:13:33.05 STC08573 00000210 BPXP0231 THREAD 2129870000000001, IN PROCESS 16974063, WAS 719
      719 00000210 TERMINATED BY SIGNAL SIGKILL, SENT FROM THREAD
      719 00000210 2128440000000002, IN PROCESS 16973917, UID 40312, IN JOB HA2ADM.
COH2 12264 15:13:33.09 STC08497 00000000 AOF571I 15:13:33 : SAPHA2AEN SUBSYSTEM STATUS FOR JOB SAPHA2AE IS 720
      720 00000000 ABENDING - SUBSYSTEM HAS SUFFERED A RECOVERABLE ERROR
COH2 12264 15:13:33.12 STC08497 00000000 AOF571I 15:13:33 : SAPHA2AEN SUBSYSTEM STATUS FOR JOB SAPHA2AE IS 721
      721 00000000 STOPPED - ABENDED, RESTARTOPT=NEVER SPECIFIED
COH2 12264 15:13:33.30 STC08497 00000000 AOF571I 15:13:33 : SAPHA2ACV SUBSYSTEM STATUS FOR JOB SAPHA2AV IS 722
      722 00000000 AUTODOWN - SET BY SHUTDOWN
COH2 12264 15:13:33.33 STC08497 00000000 AOF743I SHUTDOWN WILL NOT (RE)PROCESS SUBSYSTEM SAPHA2ACV AS IT IS 723
      723 00000000 AUTODOWN
COH3 12264 15:13:33.35 STC08713 00000000 AOF570I 15:13:33 : ISSUED "INGUSS JOBNAME=HA2CPAS,/bin/tcsh -c 901
      901 00000000 '/bin/cp -p ~/start_ASCS20_srv.COH3.log
      901 00000000 ~/start_ASCS20_srv.COH3.log.old'" FOR SUBSYSTEM SAPHA2A_SRV -
      901 00000000 MSGTYPE IS PRESTART
COH3 12264 15:13:33.36 STC08713 00000010 *HSAL6010A SAPHA2AEN/APL/COH2; INTERVENTION REQUIRED; BEYOND AUTOMATION
COH1 12264 15:13:33.37 STC08275 00000010 *HSAL6010A SAPHA2AEN/APL/COH2; INTERVENTION REQUIRED; BEYOND AUTOMATION
COH2 12264 15:13:33.42 STC08497 00000010 *HSAL6010A SAPHA2AEN/APL/COH2; INTERVENTION REQUIRED; BEYOND AUTOMATION
COH3 12264 15:13:33.43 STC01743 00000201 $HASP100 BPXAS ON STCINRDR
COH3 12264 15:13:33.44 STC08713 00000000 AOF571I 15:13:33 : SAPHA2A_SRV SUBSYSTEM STATUS FOR JOB SHA2ASR IS 904
      904 00000000 STARTED - STARTUP FOR SAPHA2A_SRV/APL/COH3 IN PROGRESS
COH3 12264 15:13:33.48 STC01743 00000010 $HASP373 BPXAS STARTED
COH2 12264 15:13:33.48 STC08497 00000000 AOF571I 15:13:33 : SAPHA2AMS SUBSYSTEM STATUS FOR JOB SAPHA2AM IS 726
      726 00000000 AUTOTERM - SET BY SHUTDOWN
COH3 12264 15:13:33.48 STC01743 00000210 BPXP024I BPXAS INITIATOR STARTED ON BEHALF OF JOB NETVIEW RUNNING IN
      ASID 001F
COH2 12264 15:13:33.49 STC08497 00000000 AOF571I 15:13:33 : SAPHA2AGW SUBSYSTEM STATUS FOR JOB SAPHA2AW IS 727
      727 00000000 AUTOTERM - SET BY SHUTDOWN
COH2 12264 15:13:33.49 STC08497 00000000 AOF571I 15:13:33 : SAPHA2AST SUBSYSTEM STATUS FOR JOB SHA2AST IS 728
      728 00000000 AUTOTERM - SET BY SHUTDOWN
COH3 12264 15:13:33.51 STC08713 00000000 AOF570I 15:13:33 : ISSUED "INGUSS JOBNAME=HA2ASCS,/bin/tcsh -c 908
      908 00000000 ~/start_sapsrv HA2 ASCS20 ha2ascsv SHA2ASR 9 >8
      908 00000000 ~/start_ASCS20_srv.COH3.log'" FOR SUBSYSTEM SAPHA2A_SRV - MSGTYPE
      908 00000000 IS STARTUP

```

After the failure, the resource SAPHA2AEN on COH2 has the status PROBLEM or HARDDOWN.

On COH3, all resources of SAPHA2ASCS and its corresponding SAP start service SAPHA2A_SRV are in AVAILABLE status. The enqueue replication server and its corresponding SAP start service have stopped on COH3.

When the enqueue server restarts on COH3, it reads the enqueue replication table from shared memory and rebuilds the enqueue table. Use the transaction SM12 to verify that the 10 lock entries you had generated are still in the enqueue table.

Look at the enqueue server log file (*enquelog*) to verify that the enqueue server restarted and the enqueue replication server is not running (there is no message specifying that replication is active).

Look at the developer trace file *dev_disp* to verify that the dispatcher lost its connection with the message server and reconnected later on.

The following log output shows the messages of the SAP system log (SM21) during the test interval.

System Log: Local Analysis of p570coh1v

System Log: Local Analysis of p570coh1v 2

Date : 20.09.2012

Time	Type	Nr	Clt	User	TCode	Priority	Grp	N	Text
15:13:14	DIA	003				0	R0	Z	The update dispatch info was reset
15:13:14	DP					0	Q0	I	Operating system call connect failed (error no. 79)
15:13:14	DP					0	Q0	N	Failed to send a request to the message server
15:13:14	DIA	003				0	R0	R	The update has been deactivated following a system error
15:13:14	DIA	003				0	R0	Z	The update dispatch info was reset
15:13:14	DP					0	Q0	N	Failed to send a request to the message server
15:13:14	DIA	003				0	R0	R	The update has been deactivated following a system error
15:13:22	DP					0	Q0	I	Operating system call connect failed (error no. 79)
15:13:42	DP					0	Q0	K	Connection to message server (on ha2ascsv) established
15:13:42	DP					0	Q1	C	MsgServer Hardware ID Was Determined
15:13:42	DIA	005				0	R0	T	The update was activated
15:13:42	SPO	014	000	SAPSYS		0	Q0	I	Operating system call writev failed (error no. 32)
15:13:42	SPO	014	000	SAPSYS		0	Q0	I	Operating system call connect failed (error no. 79)
15:13:48	DIA	005	000	SAPSYS		0	Q0	I	Operating system call writev failed (error no. 32)
15:14:48	DIA	003	000	SAPSYS		0	Q0	I	Operating system call writev failed (error no. 32)

Reading:
 Number of records read..... 0000003121
 Number of records selected..... 0000000015
 Old records skipped..... 0000003106
 Further selection:
 Number of records read..... 0000000015
 Number of records selected..... 0000000015
 Number of records printed..... 0000000015
 End of system log

System Log: Local Analysis of p570coh1v 3
 Contents

SM21 p570coh1v INS

Figure 46. SAP system log (SM21)

Failure of the message server

This scenario simulates the failure of the message server and tests the behavior of SA z/OS.

The following table summarizes the execution of the test.

Table 32. Failure of the message server

Scenario characteristics	Description
Purpose	Scope: Message server Action: Unplanned outage
Expected behavior	Example ABAP message server: SA z/OS waits until the message server process is automatically restarted by SAP itself. The sapstart process of the ASCS instance restarts the message server. This is caused by the Restart_Program_<xx> entry in the ASCS profile. For details see <i>SAP Note 2177923: Processes started by SAP start service are not auto-restarted when terminated due to error.</i> The short interrupt, until the message server is restarted, can have an impact to the workload, see Table 5 on page 85.
Setup	The LPARs COH1, COH2 and COH3 must be running, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> the SCSs inclusive the message server running on COH3 the enqueue replication server running on COH1.
Preparation	<ul style="list-style-type: none"> Log on to an SAP application server. Create workload (SGEN) and entries in the enqueue table.
Execution	Use the UNIX command kill once with signal -9 and once with signal -2 to kill the message server process outside of SA z/OS. Both signals have the same effect to the SA processing.
Verifications	<ul style="list-style-type: none"> 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).
Observed results, if unexpected:	

Before the test, all SAP-related resources are in AVAILABLE status. The message and enqueue servers are running on COH3, and the enqueue replication server is running on COH1.

The workload and enqueue entries are created.

Then simulate the failure of the ABAP message server process (`ms.sapHA2_ASCS20`) using the UNIX command `kill -9 <pid>`:

```

coh3vipa,~,7:15pm,1,#ps -ef | grep -i ha2
ha2adm 16908559 33686161 - 13:38:15 ? 0:13 en.sapHA2_ASCS20 pf=/usr/sap/HA2/SYS/profile/HA2_ASCS20_ha2ascsv
root 131345 131386 - 19:15:28 tty0000 0:00 grep -i ha2
ha2adm 16908570 1 - 13:38:02 ? 0:15 /usr/sap/HA2/ASCS20/exe/sapstartsrv pf=/usr/sap/HA2/SYS/profile/HA2_ASCS20_ha2
ha2adm 67240345 33686161 - 13:38:15 ? 0:05 ms.sapHA2_ASCS20 pf=/usr/sap/HA2/SYS/profile/HA2_ASCS20_ha2ascsv
ha2adm 33686161 1 - 13:38:14 ? 0:00 sapstart pf=/usr/sap/HA2/SYS/profile/HA2_ASCS20_ha2ascsv

```

```
coh3vipa,~,7:15pm,2,#kill -9 67240345
```

```

coh3vipa,~,7:20pm,3,#ps -ef | grep -i ha2
ha2adm 16908559 33686161 - 13:38:15 ? 0:14 en.sapHA2_ASCS20 pf=/usr/sap/HA2/SYS/profile/HA2_ASCS20_ha2ascsv
root 131347 131386 - 19:22:32 tty0000 0:00 grep -i ha2
ha2adm 16908570 1 - 13:38:02 ? 0:15 /usr/sap/HA2/ASCS20/exe/sapstartsrv pf=/usr/sap/HA2/SYS/profile/HA2_ASCS20_ha2
ha2adm 84017561 33686161 - 19:20:50 ? 0:00 ms.sapHA2_ASCS20 pf=/usr/sap/HA2/SYS/profile/HA2_ASCS20_ha2ascsv
ha2adm 33686161 1 - 13:38:14 ? 0:00 sapstart pf=/usr/sap/HA2/SYS/profile/HA2_ASCS20_ha2ascsv

```

The sapstart process restarts SAP message server immediately in place, on COH3.

The failure is transparent: the workload is still running (SM66), and the lock entries that were generated are still in the enqueue table (SM12).

Looking at the trace file of the dispatcher (dev_disp), verify that it lost its connection with the message server and reconnected a few seconds later.

Failure of the enqueue replication server

This scenario simulates the failure of the enqueue replication server. The following table summarizes the execution of the test.

Table 33. Failure of the enqueue replication server

Scenario characteristics	Description
Purpose	Scope: Enqueue replication server Action: Unplanned outage
Expected behavior	SA z/OS restarts the enqueue replication server in place. The failure has no impact on the SAP workload.
Setup	The LPARs COH1, COH2 and COH3 must be running, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> the enqueue server running on COH3. the enqueue replication server running on COH1.
Preparation	<ul style="list-style-type: none"> Log on to an SAP application server. Create entries in the enqueue table.
Execution	Use the UNIX command <code>kill</code> once with signal <code>-9</code> and once with signal <code>-2</code> to kill the enqueue replication server process outside of SA z/OS.
Verifications	<ul style="list-style-type: none"> 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 <code>dev_disp</code> and in the system log (SM21).
Observed results, if unexpected:	

Before the test, all SAP-related resources are in AVAILABLE status. The message- and enqueue servers are running on COH3, and the enqueue replication server is running on COH1.

As described in “Preparing for the test” on page 201, log on to all SAP application servers and generate 10 lock entries in the enqueue table.

Then simulate the failure: kill the enqueue replication server process using the UNIX command `kill -9 <pid>`. Use the `ps -ef` and `kill` commands as shown in “Failure of the message server” on page 220 to retrieve the process ID of the enqueue replication server, to stop it, and to verify that the process is restarted.

SA z/OS immediately restarted the enqueue replication server in place on COH1. After restarting, the enqueue replication server reconnects to the enqueue server and rebuilds its enqueue replication table in shared memory.

The failure is transparent: the workload is still running (SM66), and the lock entries that you generated are still in the enqueue table (SM12). Additionally, you can check the connection between the enqueue server and the enqueue replication server with the SAP utility program `ensmon`. Calling `ensmon` with option 2 (that is: Get replication information) displays:

```
ha2adm> ensmon pf=/usr/sap/HA2/SYS/profile/HA2_ASCS20_ha2ascsv 2
```

This provides the following output:

```
Try to connect to host ha2ascsv service sapdp20
get replinfo request executed successfully
Replication is enabled in server, repl. server is connected
Replication is active
...
```

Failure of the SAP start service

This scenario simulates the failure of the SAP start service process `sapstartsrv`. The following table summarizes the execution of the test.

Table 34. Failure of the SAP start service

Scenario characteristics	Description
Purpose	Scope: SAP start service of ASCS, SCS and ERS (ABAP or Java, or both). Action: Unplanned outage
Expected behavior	SA z/OS restarts the SAP start service. <ul style="list-style-type: none"> • If the service was stopped by a <code>kill -2</code> command and the instance is active, then this causes the restart of <code>sapstartsrv</code> and the complete ASCS or SCS group, or ERS group. You implicitly execute the test described in “Failure of the sapstart process of the SAP Central Services” on page 223 or “Failure of the sapstart process of the enqueue replication server” on page 225. • If the service was stopped by a <code>kill -2</code> command and the instance is inactive, then <code>sapstartsrv</code> just restarts in place. • Stopping the service by a <code>kill -9</code> command causes a restart of <code>sapstartsrv</code> in place and lets an active instance untouched. <p>The failure has no impact on the SAP workload, if <code>sapstartsrv</code> has been killed with signal -9. Also the failure is transparent, if <code>sapstartsrv</code> of the ERSs has been killed with signal -2. In both cases, the enqueue and message servers are not restarted.</p> <p>Workload can be impacted, if <code>sapstartsrv</code> of the SCS or ASCS group was stopped with <code>kill -2</code>, because then the enqueue and message servers are restarted, see Table 5 on page 85.</p>

Table 34. Failure of the SAP start service (continued)

Scenario characteristics	Description
Setup	The LPARs COH1, COH2 and COH3 must be running, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> • the ABAP enqueue server running on COH1. • the ABAP enqueue replication server running on COH2.
Preparation	<ul style="list-style-type: none"> • Log on to an SAP application server. • Create workload (SGEN) and entries in the enqueue table.
Execution	Use the UNIX command <code>kill</code> once with signal <code>-9</code> and once with signal <code>-2</code> to kill the SAP start service outside of SA z/OS. The signals have different effects on the processing of SA z/OS.
Verifications	<ul style="list-style-type: none"> • 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 <i>dev_disp</i> and in the system log (SM21). • Check the SAP System log. There should be no error about the ERS failure (SM21).
Observed results, if unexpected:	

Before the test, all SAP-related resources are in AVAILABLE status. The `sapstartsrv` and the ABAP SCS run on COH1. The workload and enqueue entries are created.

Then simulate the failure: kill the SAP start service `sapstart` of the ABAP Central Services using the UNIX command `kill -9 <pid>`. Use the `ps -ef` and `kill` commands as shown in “Failure of the message server” on page 220 to retrieve the process ID of the process, to stop it, and to verify that the process is moved or restarted.

You see that only the `sapstartsrv` of the ABAP SCS has been restarted in place. All other resources of the ABAP SCS Group are still running.

Stopping `sapstartsrv` of an inactive ASCS, SCS, or ERS with command `kill -2` causes the same processing of SA: restarting `sapstartsrv` in place.

The result of stopping `sapstartsrv` with `kill -2` of an active ASCS, SCS, or ERS is described in “Failure of the `sapstart` process of the SAP Central Services” and “Failure of the `sapstart` process of the enqueue replication server” on page 225.

Failure of the `sapstart` process of the SAP Central Services

This scenario simulates the failure of the `sapstart` process from the SAP Central Services and tests the behavior of SA z/OS. The following table summarizes the execution of the test.

Table 35. Failure of the sapstart process of the SAP Central Services

Scenario characteristics	Description
Purpose	Scope: sapstart process of ASCS and SCS, or both. Action: Unplanned outage
Expected behavior	SA z/OS shows a PROBLEM/HARDDOWN status for the failed resource SAPHA2AST and restarts the ABAP Central Services (SAPHA2ASCS) on the LPAR, where the enqueue replication server was active before the test. Before the restart, its corresponding SAP start service (SAPHA2A_SRV) moves, because of the <i>HasParent</i> relationship to SAPHA2ASCS. The enqueue replication server stops and restarts on a free LPAR. Before the restart, its corresponding SAP start service moves to the free LPAR, because of the <i>HasParent</i> relationship to SAPHA2AERS or SAPHA2JERS. The short interrupt, until the message server is started on the other LPAR, can have an impact to the workload, see Table 5 on page 85.
Setup	The LPARs COH1, COH2 and COH3 must be running, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> • the ABAP SAP Central Services running on COH1. • the ABAP enqueue replication server running on COH2.
Preparation	<ul style="list-style-type: none"> • Log on to an SAP application server. • Create workload (SGEN) and entries in the enqueue table.
Execution	Use the UNIX command <code>kill</code> once with signal <code>-9</code> and once with signal <code>-2</code> to kill the SAP start service outside of SA z/OS. Both signals result in the same SA z/OS reaction.
Verifications	<ul style="list-style-type: none"> • 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 <i>dev_disp</i> and in the system log (SM21). • Check in the System log, when the SAP application server lost and re-established the connection to the message server (SM21).
Observed results, if unexpected:	

Before the test, all SAP-related resources are in AVAILABLE status. The enqueue servers are running on COH1, and the enqueue replication server is running on COH2. Workload and enqueue entries are created.

Then simulate the failure: kill the SAP start service sapstart of the ABAP SAP Central Services using the UNIX command `kill -9 <pid>`. Use the `ps -ef` and `kill` commands as shown in “Failure of the message server” on page 220 to retrieve the process ID of the sapstart process, to stop it, and to verify that the ASCS instance has moved as well as its corresponding ERS instance.

You will see that the sapstart process of the ABAP SAP central service has been restarted on LPAR COH2 (where ERS is running). All other SAP resources of the ABAP SAP central service Group have been restarted as well on LPAR COH2 at the same time. The ERS for ABAP was moved from LPAR COH2 to COH3.

Note: After the failure, the resource SAPHA2AST on COH1 has the SA z/OS agent status PROBLEM/HARDDOWN. With this status, the ASCS does not restart on COH1. To make COH1 available again for ASCS, you must update the SA z/OS agent status to AUTODOWN for SAPHA2AST on COH1.

In a productive environment, you should always verify the cause of the process failure, before you change the agent status from HARDDOWN to AUTODOWN.

Failure of the sapstart process of the enqueue replication server

This scenario simulates the failure of the sapstart process from the SAP Central Services and tests the behavior of SA z/OS. The following table summarizes the execution of the test.

Table 36. Failure of the sapstart process of the enqueue replication server

Scenario characteristics	Description
Purpose	Scope: sapstart process of ABAP or Java ERS or both Action: Unplanned outage
Expected behavior	SA z/OS restarts the sapstart process in place. <ul style="list-style-type: none"> • If the service was stopped by a kill -2 command, then the enqueue replication server is also restarted in place. • If the service was stopped by a kill -9 command, then SA z/OS does neither restart the sapstart process, nor the enqueue replication server itself. <p>The failure has no impact on the SAP workload.</p>
Setup	The LPARs COH1, COH2 and COH3 must be running, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> • the ABAP SAP Central Services running on COH1 • the ABAP enqueue replication server running on COH2.
Preparation	<ul style="list-style-type: none"> • Log on to all SAP application servers. • Create entries in the enqueue table.
Execution	Use the UNIX command kill once with signal -9 and once with signal -2 to kill the SAP start service process outside of SA z/OS. The signals have different effects on the processing of System Automation.
Verifications	<ul style="list-style-type: none"> • 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 <i>dev_disp</i> and in the system log (SM21).
Observed results, if unexpected:	

Before the test, all SAP-related resources are in AVAILABLE status. The ABAP enqueue replication server is running on COH3. The enqueue entries are created.

Then simulate the failure: kill the SAP start service sapstart of the ABAP Central Services using the UNIX command kill -2 <pid>. Use the ps -ef and kill commands as shown in “Failure of the message server” on page 220 to retrieve the process ID of the sapstart process, to stop it, and to verify that the ERS instance was restarted in place.

You will see that the sapstart process of the ABAP ERS has been restarted on the same LPAR COH3. Also the enqueue replication server process has been restarted in place at the same time.

Note: If the sapstart process of the ERS has been killed with signal -9, then the sapstart process is not restarted by SA z/OS. This is because the sapcontrol GetProcessList command, which checks if the instance is running OK, returns no errors (rc=3), even without the sapstart process running. The sapstart process is indeed not needed for running the ERS process properly. To clear this situations, perform the following steps:

1. additionally, kill the ERS process with signal -9. It can take up to 30 seconds until the ERS status changes into status HARDDOWN.
2. Change (for example, in SA z/OS) the agent status from HARDDOWN into AUTODOWN. Then the ERS instance restarts including the sapstart process on the same LPAR.

Failure of the NFS server

This scenario simulates the failure of the NFS server and tests the behavior of SA z/OS. It also measures the impact of the failure on the SAP workload.

The following table summarizes the execution of the test.

Table 37. Failure of the NFS server

Scenario characteristics	Description
Purpose	Scope: NFS server Action: Unplanned outage
Expected behavior	SA z/OS should restart the NFS server. Existing NFS mounts should be reestablished. The global file systems can be accessed from each SAP application server
Setup	COH1, COH2 and COH3 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> • The NFS server running on COH1
Preparation	<ul style="list-style-type: none"> • Log on to all SAP application servers.
Execution	Cancel the address space MVSNFSSA on COH1.
Verifications	<ul style="list-style-type: none"> • Check that the file systems are accessible (AL11). • Look for error messages in the system log (SM21).
Observed results, if unexpected:	

Before the test, all SAP-related resources are in AVAILABLE status. The NFS and enqueue servers are running on COH1.

Simulate the failure by canceling the address space of the NFS server on COH1 using the following command:

```
/C MVSNFSSA
```

Because, at the time of the test, the effective preference of COH2 was higher than that of COH3, SA z/OS immediately restarted the NFS server on COH2 (along with its VIPA) and put the resource MVSNFSSA on COH3 in a RESTART status:

```

AOFKSTA5          SA z/OS - Command Dialogs          Line 1 of 3
Domain ID = IPXFO ----- DISPSTAT -----         Date = 03/30/10
Operator ID = HEIKES                                  Time = 15:37:01
 A dispflgs 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 T RS TYPE   Activity
-----
NFSSERV      DOWN     COH1     MVSNFSHA - - - - - MVS   --none--
NFSSERV      UP       COH2     MVSNFSHA - - - - - MVS   --none--
NFSSERV      RESTART COH3     MVSNFSHA - - - - - MVS   --none--

```

The SAP global file systems that are NFS-mounted on AIX box p570coh2 are accessible with SAP transaction AL11. No error messages are written to the SAP system log (SM21).

Failure of a TCP/IP stack

This scenario simulates the failure of the TCP/IP stack on the system where the enqueue server and the NFS server are running, and tests the behavior of SA z/OS. It also measures the impact of the failure on the SAP workload.

The samples from the scenario in this section use a TCPIP stack name of TCPIPA.

The following table summarizes the execution of the test.

Table 38. Failure of a TCP/IP stack

Scenario characteristics	Description
Purpose	Scope: TCP/IP stack Action: Unplanned outage
Expected behavior	SA z/OS restarts the TCP/IP stack. OMPROUTE goes into a PROBLEM/HARDDOWN status, and therefore the NFS server fails and SA z/OS moves it. Because of the <i>HasParent</i> relationship definition in the SAP policy, SA z/OS stops SCS and restarts it on the LPAR where the enqueue replication server is running. As a consequence, the enqueue replication server starts on a different LPAR, as well due to the <i>HasParent</i> relationship definition in the SAP policy. For the remote SAP 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 SAP application server running on the other LPAR, the failure should have no impact.

Table 38. Failure of a TCP/IP stack (continued)

Scenario characteristics	Description
Setup	LPARs COH1, COH2 and COH3 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 COH1 The enqueue replication server running on COH2 The NFS server running on COH1
Preparation	<ul style="list-style-type: none"> Log on to an SAP application, which is connected to a DB2 subsystem on COH1. Create workload with client copy.
Execution	Cancel the address space TCPIPA on COH1.
Verifications	<ul style="list-style-type: none"> Check if the workload is still running (SM50/SM66). Look for error messages in the enqueue log file, in the developer traces <i>dev_disp</i> and <i>dev_w<x></i>, where x is the number of the work process it belongs to, and in the system log (SM21).
Observed results, if unexpected:	

Before the test, all SAP-related resources are in AVAILABLE status. The NFS and the SCS (with enqueue- and message servers) are running on COH1, and the enqueue replication server runs on COH2.

Simulate the failure by stopping TCPIPA on COH1 using the following command:
/C TCPIPA

Because the critical threshold is not reached, SA z/OS immediately restarts TCPIPA on COH1.

The failure of the TCP/IP stack leads to the failure of the NFS server and SCS on COH1.

SA z/OS immediately restarts the NFS server on another LPAR (COH2).

SA z/OS restarts SCS on the LPAR where the enqueue replication server is running, that is COH2. The enqueue replication server was restarted on COH3:

```

INGKYST0          SA z/OS - Command Dialogs          Line 1    of 18
Domain ID = IPXFO          ----- INGLIST -----          Date = 03/05/10
Operator ID = HEIKES          Sysplex = COHPLEX          Time = 14:00:25
A Update  B Start  C Stop   D INGRELS  E INGVOTE  F INGINFO  G Members
H DISPTRG I INGSCHED J INGGROUP K INGCICS  L INGIMS   M DISPMTR  T INGTWS
U User    X INGLKUP / scroll

CMD Name      Type System  Compound  Desired  Observed  Nature
-----
SAPHA2AER     APL COH1    SATISFACTORY UNAVAILABLE SOFTDOWN
SAPHA2AER     APL COH2    SATISFACTORY UNAVAILABLE SOFTDOWN
SAPHA2AER     APL COH3    SATISFACTORY AVAILABLE    AVAILABLE
SAPHA2JER     APL COH1    SATISFACTORY UNAVAILABLE SOFTDOWN
SAPHA2JER     APL COH2    SATISFACTORY UNAVAILABLE SOFTDOWN
SAPHA2JER     APL COH3    SATISFACTORY AVAILABLE    AVAILABLE
    
```

The client copy is started interactively (not as background job). ClientCopy fails because it can not handle the TCPIP failure transparently. After TCPIP is up again

and the SCS and enqueue replication server are restarted, the client copy can be restarted and finishes without errors and without lock entries in the lock table (can be checked with SAP transaction SM12).

The transaction DB2 shows that the current DB host is still ihlscoh1. Use the DB2 command `-DIS THREAD(*)` to check that all the threads are connected to COH1. Connection information for each work process can be found in the developer trace file `dev_w<x>`, where `x` is the number of the work process it belongs to.

The developer trace `dev_disp` shows that the dispatcher lost its connection with the message server and reconnected later on.

The SAP system log (SM21) shows error and recovery messages issued during the interval of the test, similar to the ones shown in Figure 46 on page 219.

All the SAP-related resources are in AVAILABLE status after the failover. The NFS and enqueue servers are running on COH2. The enqueue replication server is running on COH3.

Failure of an LPAR

This scenario simulates the failure of the LPAR where the enqueue server and the NFS server were running and test the behavior of SA z/OS. It also measures the impact of the failure on the SAP workload.

Table 39 summarizes the execution of the test.

Table 39. Failure of the LPAR where the ES and NFS servers are running

Scenario characteristics	Description
Purpose	Scope: One LPAR Action: Unplanned outage
Expected behavior	SA z/OS should restart the master of the failing DB2 subsystem on another LPAR in light mode. The DB2 subsystem goes down after successful startup. SA z/OS should restart the NFS server on another LPAR. SA z/OS should restart SCS 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. For the SAP application server connected to the database server which is running on the failing LPAR, running transactions should be rolled back and work processes should failover to the standby database server. For the SAP application server connected to that database server which is running on the other LPAR, the failure should have no impact.
Setup	The LPARs COH1, COH2 and COH3 must be up, including all required z/OS resources and SAP-related resources, with: <ul style="list-style-type: none"> • The SCS with enqueue server running on COH2. • The enqueue replication server running on another LPAR (COH3).

Table 39. Failure of the LPAR where the ES and NFS servers are running (continued)

Scenario characteristics	Description
Preparation	<ul style="list-style-type: none"> Log on to an SAP application server, which is connected to database server on COH2. Create a workload with client copy on the application server which is connected to the database of COH2.
Execution	System reset at the HMC for COH2 (PAM in SA z/OS).
Verifications	<ul style="list-style-type: none"> Check if the workload is still running (SM50, SM66). Verify the number of entries in the enqueue table (SM12). Look for error messages in the enqueue log file, in the developer traces <i>dev_disp</i> and <i>dev_w<x></i>, where x is the number of the work process it belongs to, and in the system log (SM21).
Observed results, if unexpected:	

Before the test, all SAP-related resources are in UP status. The NFS was started on COH1. The SCS (ABAP and Java) started on COH3 (the LPAR where enqueue replication server was running before) and the enqueue replication server is running on COH1.

Simulate the failure by doing a system reset at the HMC.

Use the SA z/OS command INGLIST */*/COH2 to display the status of the resources on COH2. They are all displayed with a status INHIBITED/SYSGONE.

SA z/OS restarts the DB2 subsystem SNH2 on COH1 with the in light mode in order to quickly release the retained locks. When the LPAR startup was complete, the DB2 subsystem in light mode stops and the DB2 primary subsystem on COH2 starts.

SA z/OS restarts the NFS server on COH1.

SA z/OS restarts SCS on the LPAR COH1, where the enqueue replication server is running.

SA z/OS restarts the enqueue replication server on COH3.

The transaction DB2 showed that the current DB host was now coh1vipa. Check with the DB2 command -DIS THREAD(*), that all the threads are connected to COH1. Connection information for each work process can be found in the developer trace files.

All SAP-related resources are in AVAILABLE status after the failover and running on COH2, including the enqueue servers. The enqueue replication server was running on COH3.

Problem determination methodology

This topic describes how to perform problem determination SA z/OS and for each of the critical SAP components.

This topic contains these main subtopics:

- “SA z/OS problem determination”
- “Where to check for application problems” on page 234
- “Checking the network” on page 236
- “Checking the status of zFS file systems and NFS” on page 238
- “Checking the status of DB2 and SAP connections” on page 239

SA z/OS problem determination

SAP HA is a complex environment, and in such an environment problems can occur. This topic directs you to areas where you can check for problems if you encounter various errors.

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 wrap-around 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.

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.

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 SA z/OS, because 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:

```
.NO_ENTRY,SUP(NO),RETAIN(I,CE),AUTO(YES)
.DEFAULT,SUP(YES),RETAIN(I,CE),AUTO(NO)
/*
/* MESSAGE SUPPRESSION
$HASP000                OK FROM JES2
$HASP100
ICH70001I
IEF196I                JOBLOG FOR SUB=MSTR
IKT005I
/*
/* AUTOMATION INITIALIZATION
AOF603D,SUP(NO),RETAIN(NO),AUTO(YES)
/*
/* DO NOT REROUTE AOF-MESSAGES BACK TO NETVIEW WHERE THEY COME FROM
AOF*,SUP(NO),RETAIN(NO),AUTO(NO)
```

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.

Problem determination in SA z/OS

Problem determination in SA z/OS really depends on the kind of error you encounter, but you should check these areas for indications:

- SDF or NMC (NetView Management Console)
- 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 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

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

Reaching the critical threshold of a resource during test activities

While simulating SAP resource outages during testing you may reach the *critical threshold value* for a resource.

The resource goes into a hard down state. In the INGINFO or DISPINFO panels for a resource in hard down status, a message is displayed, such as CRITICAL THRESHOLD EXCEEDED.

Before you can perform further tests you must reset the *critical threshold value* for a resource. To do this, issue the following command:

```
ASF REQ=REPL,ID=<Resource>,ERRORDT=''
```

For information about the ASF command, refer to *Tivoli System Automation: Operator's Commands*.

UNIX messages

By default, UNIX messages are not 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
```

In some cases you might need a more detailed explanation for the failure of certain z/OS UNIX commands that you issue from your SA z/OS policy. The `errno2` code is reported if you set the variable `_EDC_ADD_ERRNO2` in the environment where you issue the z/OS UNIX command. You find a listing of these `errno2` values in *z/OS UNIX System Services Messages and Codes* in topic *z/OS UNIX reason codes*.

The UNIX messages appear in the z/OS syslog with a BPXF024I message ID. To send them further to NetView, you might have to modify MPF (see “Message Processing Facility” on page 232).

If nothing happens

You may encounter a failure situation in which you enter a command to SA z/OS and nothing happens; there is no error message, and there are no status changes shown on SDF or NMC.

Typically this situation occurs because there is a lock in the system, which can have various causes. This section describes these causes and shows how you can determine the reason of the problem:

- 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?
- Auto flags in the SA z/OS agent
 - Enter: *DISPSTAT application name* and examine the automation flags.
- 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 flag.

When you are really lost

The last step before calling IBM support could be to do a cold start of the automation manager (INGEAMSA). A cold start will usually get rid of possible deadlocks, but note the following caveat.

Important: An automation manager cold start also deletes 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 INGEAMSA, 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 INGEAMSA,sub=mstr,type=cold
```

After the primary automation manager initializes, start the secondary automation managers.

Where to check for application problems

This information describes where to look if SA z/OS 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 SA z/OS, 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/<sid>adm.

The following command shows the log files in chronological order:

```
ls -rtl *.log
```

- Log file does not exist.

In this case, SA z/OS 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 231).
- Check the z/OS UNIX system log (*syslogd*) for messages.
- Check the availability of file systems. Are the SAP global, profile, and exe directories accessible?
- Logon to z/OS UNIX and execute the command manually.
- For remote resources, the log files usually indicate the reason that SA z/OS 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.

- **SAP enqueue server, message server, gateway does not start**

First check, if the `sapstart.log` file in the work directory of the SCS instance is correctly tagged. If it is incorrectly tagged, then the SCS Instance does not start under SA z/OS, because it cannot be read. See an example, where it is incorrectly tagged:

```
| coh1vipa:/usr/sap/HA2/SCS21/work (2)>ls -rtlT sapstart.log  
| t IBM-1047 T=on -rw-rw-rw- 1 ha2adm sapsys 2525 Mar 6 15:10 sapstart.log
```

This is the correct tagging:

```
| coh1vipa:/usr/sap/HA2/SCS21/work (3)>ls -rtlT sapstart.log  
| - untagged T=off -rw-r--r-- 1 ha2adm sapsys 1990 Jul 6 09:33 sapstart.log
```

To fix this:

1. Rename or delete `sapstart.log`.
2. Make sure that the `sapstart` executable has the minimum required patch level (see “Software prerequisites” on page 90).

To check this, logon as user `<sid>adm` under z/OS UNIX in its home directory and perform the following:

```
sapcontrol -nr <instance_nr> -host <virt_hostname> -function GetVersionInfo
```

Check messages in the log file that you defined for the SA z/OS resource in the SA z/OS policy. The log files are located in the home directory of the administrator user, for example, `/u/ha2adm`, and contain the LPAR name in their name. If the enqueue server was started on the COH1 LPAR, the policy creates the log files:

- `start_ASCS00_srv.COH1.log.old` (log file from previous start of the `sapstartsrv` service)
- `start_ASCS00_srv.COH1.log` (log file from start of the `sapstartsrv` service)
- `start_ASCS00.COH1.log.old` (log file from previous start of the ASCS instance)
- `start_ASCS00.COH1.log` (log file from start of the ASCS instance)

Further SAP log files can be found in the work directory for the central services instance, for example:

```
/usr/sap/HA2/ASCS20/work (for the SAP System NW 7.10 test system HA2)
```

For the enqueue server, browse the `enque.log` file in the work directory. It shows when the enqueue server has been started and stopped, and whether the enqueue replication server is activated.

- **The application servers do not connect to the message server or the enqueue server**

- Check the network and the routing; refer to “Checking the network.”
- Check that the enqueue server can be reached. For this purpose use the `ensmon` command:

```
ensmon -H <hostname> -I <enq_instance_number> 1
```

In this configuration, the command looks as follows:

```
ensmon -H ha2ascsv -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.

Checking the network

Describing how to troubleshoot network problems could probably fill an entire volume. Therefore, here you find a selection of useful commands to verify the configuration and the connectivity between the systems. You also find a list of 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 z/OS UNIX commands.

Checking the configuration

First, check the setup. The following command performs a basic consistency check:

```
TSO: HOMETEST
```

The following commands display the network configuration and attributes.

```
OPER: D TCPIP,,N,CONFIG
```

```
TSO: NETSTAT CONFIG
```

```
z/OS UNIX: netstat -f
```

These commands allow you to verify your specifications in the TCP/IP profile. In particular, check the following settings:

- FORWARDING YES
- IGREDIRECT 1
- SOURCEVIP 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
```

```
z/OS UNIX: netstat -d
```

From these 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 z/OS UNIX environment, use the following command to display the list of home addresses (inclusive the VIPAs):

```
z/OS UNIX: netstat -h
```

or just the dynamic VIPAs on the system:

```
z/OS UNIX: netstat -v
```

Routing tables and OSPF

To display routing tables:

```
OPER: D TCPIP,,N,ROUTE  
TSO: NETSTAT ROUTE  
z/OS UNIX: netstat -r
```

To display gateways, you can use:

```
TSO: NETSTAT GATE  
z/OS UNIX: 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
```

```
z/OS UNIX: 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 SOURCEVIPA 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).

Checking the status of zFS file systems and NFS

There are various commands with which you can check the status and further attributes of a zFS file system.

You can check the attributes of a zFS file system with the following command:

```
df -kv <filename>
```

See an example of the output: The z/OS UNIX file system ownership is on COH3 and the file system is movable:

```
ha2adm> df -kv /usr/sap/HA2
Mounted on      Filesystem                Avail/Total   Files      Status
/usr/sap/HA2   (OMVS.ZFS.COHPLEX.HA2.USRSAP) 203197/1024560 4294966793 Available
ZFS, Read/Write, Device:24, ACLS=Y
File System Owner : COH3      Automove=Y    Client=Y
Filetag : T=off  codeset=0
Aggregate Name : OMVS.ZFS.COHPLEX.HA2.USRSAP
```

To verify, whether a file system is defined sysplex-aware, use the following command:

```
zfsadm agrinfo -long -aggregate OMVS.ZFS.COHPLEX.HA2.USRSAP
```

For a sysplex-aware file system, the command output is similar to the following:

```
OMVS.ZFS.COHPLEX.HA2.USRSAP (R/W COMP): 13626955 K free out of total 14353200
version 1.5
auditfid C3D6C8D6 D7C50E14 0000
sysplex-aware
      1703355 free 8k blocks;          115 free 1K fragments
      10248 K log file;              72 K filesystem table
      1992 K bitmap file
```

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 data set 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: Subsequently, all clients must remount this NFS file system .

Checking the status of DB2 and SAP connections

This information discusses basic techniques for identifying problems that are related to the SAP connections to DB2, or DB2 itself. It describes various commands with which you can check the status of DB2 and SAP connections.

More problem determination information can be found in the *Database Administration Guide: SAP on IBM DB2 for z/OS* and the respective *Planning Guide for SAP on IBM DB2 for z/OS*.

Check that DB2 is running

Use the SDSF DA command to show the status of DB2.

The following figure shows a sample output. If the display does not show the DB2 systems running, then check the z/OS system log for messages (refer to “z/OS syslog” on page 231).

```

SDSF DA COH1 (ALL) PAG 0 CPU/L/Z 6/ 2/ 0 LINE 1-13 (13)
COMMAND INPUT ==>>> SCROLL ==>> CSR
NP JOBNAME StepName ProcStep JobID Owner C Pos DP Real Paging SIO
SNH1DIST SNH1DIST IEFPROC STC03201 SNHUSER NS FE 3788 0.00 0.00
SNH1MSTR SNH1MSTR IEFPROC STC03187 SNHUSER NS FE 1632 0.00 0.00
SNH1DBM1 SNH1DBM1 IEFPROC STC03198 SNHUSER NS FE 44T 0.00 0.00
SNH1IRLM SNH1IRLM IRLM STC03194 SNHUSER NS FE 18T 0.00 0.00
SNH2MSTR SNH2MSTR IEFPROC STC03188 SNHUSER NS FE 589 0.00 0.00
SNH2IRLM SNH2IRLM IRLM STC03195 SNHUSER NS FE 18T 0.00 0.00
SNH2DIST SNH2DIST IEFPROC STC03203 SNHUSER NS FE 1312 0.00 0.00
SNH2DBM1 SNH2DBM1 IEFPROC STC03199 SNHUSER NS FE 12T 0.00 0.00
SNH3DBM1 SNH3DBM1 IEFPROC STC03197 SNHUSER NS FE 33T 0.00 0.00
SNH3IRLM SNH3IRLM IRLM STC03192 SNHUSER NS FE 20T 0.00 0.00
SNH3DIST SNH3DIST IEFPROC STC03202 SNHUSER NS FE 1749 0.00 0.00
SNH3MSTR SNH3MSTR IEFPROC STC03186 SNHUSER NS FE 2257 0.00 0.00

```

Figure 47. 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 z/OS UNIX. For example:

```
-SNH1 DISPLAY THREAD(*)
```

The following figure shows the results of this command for configuration. It shows one AIX application server p570coh1 with ABAP and Java database connections to the DB2 member. In the example, ABAP connections use the AUTHID R3USER whereas Java connections use the AUTHID SAPJAVA.

```

DSNV402I  -SNH1 ACTIVE THREADS - 237
NAME      ST A  REQ ID      AUTHID  PLAN    ASID  TOKEN
SERVER   RA * 124 db2jcc_appli SAPJAVA  DISTSERV 0076  773
V437-WORKSTATION=p570coh1.boebli, USERID=SAPJAVA,
APPLICATION NAME=db2jcc_application
V445-GA6504EF.07EC.C66C67FB88FF=773 ACCESSING DATA FOR
::FFFF:10.101.4.239
SERVER   RA * 500 db2jcc_appli SAPJAVA  DISTSERV 0076  771
V437-WORKSTATION=p570coh1.boebli, USERID=SAPJAVA,
APPLICATION NAME=db2jcc_application
V445-GA6504EF.076E.C66C65371E12=771 ACCESSING DATA FOR
::FFFF:10.101.4.239
SERVER   RA * 291 HA2DIA003   R3USER  DISTSERV 0076  700
V437-WORKSTATION=p570coh1, USERID=r3user,
APPLICATION NAME=HA2DIA003
V442-CRTKN=10.101.4.239.33093.100813131913
V445-GA6504EF.0145.100813131913=700 ACCESSING DATA FOR
::FFFF:10.101.4.239
SERVER   RA * 599 HA2SP0014   R3USER  DISTSERV 0076  701
V437-WORKSTATION=p570coh1, USERID=r3user,
APPLICATION NAME=HA2SP0014
V442-CRTKN=10.101.4.239.33094.100813131913
V445-GA6504EF.0146.100813131913=701 ACCESSING DATA FOR
::FFFF:10.101.4.239
...
...

```

Figure 48. Results of DB2 Display Thread command

Chapter 11. Verifying your System Automation for Multiplatforms implementation on Linux or AIX

This information unit describes procedures and failover scenarios for verifying your implementation on Linux or AIX.

Previous editions of this publication described in detail how to verify an SA MP policy for SAP that was created from the sample policy, which was contained in previous versions of `zSAP_BusinessContinuity.zip`. Refer to previous editions if you continue to use this policy.

For new installations it is recommended to use the SA MP policy that is based on the *SAP high availability policy feature* of System Automation for Multiplatforms 4.1.0.2 or higher (see Chapter Customizing System Automation for Multiplatforms). Verification procedures for this policy are described in detail in the verification topic for the *SAP high availability policy feature* in *Tivoli System Automation for Multiplatforms Version 4 Release 1 High Availability Policies Guide* (SC34-2660), which is available from the *IBM Knowledge Center for Tivoli System Automation for Multiplatforms*.

Make sure you cover all scenarios in your tests - planned outages (normal operation, maintenance) and unplanned outages (failures). Each scenario should be verified for proper operation.

Chapter 12. Operating an SAP system under System Automation control

This information describes why and how to use System Automation mechanisms and commands to perform daily operations and to manage changes to your SAP system.

An SAP system that is set up according to the guidelines in this documentation has System Automation products in place that start, stop, and control SAP resources, as well as the infrastructure resources needed by SAP.

The benefits of automation are that, in contrast to a manually operated SAP system, standard operational tasks such as starting and stopping parts or the entire SAP system, are run via the automation software. There is usually no requirement for you to intervene manually in normal SAP system operations, in fact it is *not* recommended that you issue manual start or stop commands against resources which are under automation control.

An administrator of such an automated SAP system must be aware that automation is active and use the mechanisms provided by the automation software to act on the resources.

The degree to which you have automated your SAP systems may vary and depends on the automation option that you have implemented (see “How to decide which System Automation option to implement” on page 24). Depending on the option you have selected, the automation products comprise Tivoli System Automation for z/OS (SA z/OS), and may include Tivoli System Automation for Multiplatforms.

Managing an SAP system

This topic describes the basic administrator operations for managing an SAP system, for example, how to start and stop an SAP system using various automation products. It shows how Tivoli System Automation products can help to make it easy to perform these basic operations.

Note: The following instructions assume that your system automation products run with the policies that are described in this publication.

A highly available SAP system includes two or more SAP application servers, SAP Central Services and a DB2 database on z/OS. There are dependencies between these SAP components and you must start and stop them in a given order.

Stopping an SAP system with SA z/OS

In an SAP installation where the SAP application servers are managed remotely from z/OS, you can use SA z/OS to stop or start the SAP system. You might use the NetView command session (SA z/OS command dialog) to control SA z/OS resources.

To stop an SAP system, you must follow a specific sequence:

- Stop the SAP application servers.
- Stop SAP Central Services Java.

- Stop SAP Central Services ABAP.
- Stop DB2.

The following procedure describes how to stop the SAP system in more detail. The samples show the commands for a dual-stack (ABAP and Java) SAP system HA4.

Note: Newer SAP NetWeaver systems are only installed as single-stack systems.

Stopping the SAP application servers

The SAP application servers are remotely managed and controlled from SA z/OS. In the NetView command session, issue INGLIST SAP<SID>R* to view all resources for all remotely managed SAP application servers of a specific SID. For SID HA4 the screen looks like shown in Figure 49.

```

INGKYST0          SA z/OS - Command Dialogs          Line 1    of 21
Domain ID = IPXFO  ----- INGLIST -----         Date = 08/22/13
Operator ID = RHIMM          Sysplex = COHPLEX          Time = 07:24:34
A Update B Start C Stop D INGRELS E INGVOTE F INGINFO G Members
H DISPTRG I INGSCHED J INGGROUP K INGCICS L INGIMS M DISPMTR T INGTWS
U User X INGLKUP / scroll
  
```

CMD Name	Type	System	Compound	Desired	Observed	Nature	Automation	Startable	Health	Auto	Hold
SAPHA4RAS_X	APG			SATISFACTORY	AVAILABLE	AVAILABLE	SERVER	INTERNAL	YES	N/A	YES NO
SAPHA4RA0	APL	COH1		SATISFACTORY	UNAVAILABLE	SOFTDOWN		IDLE	YES	N/A	YES NO
SAPHA4RA0	APL	COH2		SATISFACTORY	AVAILABLE	AVAILABLE		IDLE	YES	N/A	YES NO
SAPHA4RA0	APL	COH3		SATISFACTORY	UNAVAILABLE	SOFTDOWN		IDLE	YES	N/A	YES NO
SAPHA4RA1	APL	COH1		SATISFACTORY	UNAVAILABLE	SOFTDOWN		IDLE	YES	N/A	YES NO
SAPHA4RA1	APL	COH2		SATISFACTORY	AVAILABLE	AVAILABLE		IDLE	YES	N/A	YES NO
SAPHA4RA1	APL	COH3		SATISFACTORY	UNAVAILABLE	SOFTDOWN		IDLE	YES	N/A	YES NO
SAPHA4RJ0	APL	COH1		SATISFACTORY	UNAVAILABLE	SOFTDOWN		IDLE	YES	N/A	YES NO
SAPHA4RJ0	APL	COH2		SATISFACTORY	AVAILABLE	AVAILABLE		IDLE	YES	N/A	YES NO
SAPHA4RJ0	APL	COH3		SATISFACTORY	UNAVAILABLE	SOFTDOWN		IDLE	YES	N/A	YES NO
SAPHA4RJ1	APL	COH1		SATISFACTORY	UNAVAILABLE	SOFTDOWN		IDLE	YES	N/A	YES NO
SAPHA4RJ1	APL	COH2		SATISFACTORY	AVAILABLE	AVAILABLE		IDLE	YES	N/A	YES NO
SAPHA4RJ1	APL	COH3		SATISFACTORY	UNAVAILABLE	SOFTDOWN		IDLE	YES	N/A	YES NO
SAPHA4RM0_X	APG			SATISFACTORY	AVAILABLE	AVAILABLE	MOVE	INTERNAL	YES	N/A	YES NO
SAPHA4RM1_X	APG			SATISFACTORY	AVAILABLE	AVAILABLE	MOVE	INTERNAL	YES	N/A	YES NO
SAPHA4R0	APG	COH1		SATISFACTORY	UNAVAILABLE	SOFTDOWN	SERVER	INTERNAL	YES	N/A	YES NO
SAPHA4R0	APG	COH2		SATISFACTORY	AVAILABLE	AVAILABLE	SERVER	INTERNAL	YES	N/A	YES NO
SAPHA4R0	APG	COH3		SATISFACTORY	UNAVAILABLE	SOFTDOWN	SERVER	INTERNAL	YES	N/A	YES NO
SAPHA4R1	APG	COH1		SATISFACTORY	UNAVAILABLE	SOFTDOWN	SERVER	INTERNAL	YES	N/A	YES NO
SAPHA4R1	APG	COH2		SATISFACTORY	AVAILABLE	AVAILABLE	SERVER	INTERNAL	YES	N/A	YES NO
SAPHA4R1	APG	COH3		SATISFACTORY	UNAVAILABLE	SOFTDOWN	SERVER	INTERNAL	YES	N/A	YES NO

Figure 49. NetView command session display for stopping SAP application servers (1)

To stop all SAP application server resources for SID HA4, set a STOP vote for the SAP HA4 remote application server group SAPHA4RAS_X. Type character c into the command column in front of SAPHA4RAS and hit the Enter key. On the next screen, you see a couple of stop options. Look for option *Type*. The default for *Type* is NORM. Change to FORCE only if you do not have changed or set up the remote application server policy to use the REXX script SAPRASTP in SHUTNORM. Hit the enter key. Verify the affected resources and hit the PF10 key to actually initiate the stop sequence.

Note: Without using REXX script SAPRASTP, a stop with TYPE=NORM (default) stops the z/OS monitoring of the remote SAP AS only. The SAP application server itself continues to run.

Figure 50 on page 245 an INGLIST panel where all SAP application server resources are stopped.

```

INGKYST0          SA z/OS - Command Dialogs      Line 1    of 21
Domain ID = IPXFO          ----- INGLIST -----      Date = 08/22/13
Operator ID = RHIMM          Sysplex = COHPLEX          Time = 07:31:55
A Update  B Start  C Stop  D INGRELS  E INGVOTE  F INGINFO  G Members
H DISPTRG I INGSCHED J INGGROUP K INGCICS  L INGIMS  M DISPMTR  T INGTWS
U User    X INGLKUP / scroll

CMD Name      Type System  Compound      Desired      Observed      Nature      Automation Startable  Health      Auto Hold
-----
SAPHA4RAS_X  APG          SATISFACTORY UNAVAILABLE  SOFTDOWN     SERVER      INTERNAL   YES      N/A      YES NO
SAPHA4RA0    APL COH1      SATISFACTORY UNAVAILABLE  SOFTDOWN     IDLE        YES      N/A      YES NO
SAPHA4RA0    APL COH2      SATISFACTORY UNAVAILABLE  SOFTDOWN     IDLE        YES      N/A      YES NO
SAPHA4RA0    APL COH3      SATISFACTORY UNAVAILABLE  SOFTDOWN     IDLE        YES      N/A      YES NO
SAPHA4RA1    APL COH1      SATISFACTORY UNAVAILABLE  SOFTDOWN     IDLE        YES      N/A      YES NO
SAPHA4RA1    APL COH2      SATISFACTORY UNAVAILABLE  SOFTDOWN     IDLE        YES      N/A      YES NO
SAPHA4RA1    APL COH3      SATISFACTORY UNAVAILABLE  SOFTDOWN     IDLE        YES      N/A      YES NO
SAPHA4RJ0    APL COH1      SATISFACTORY UNAVAILABLE  SOFTDOWN     IDLE        YES      N/A      YES NO
SAPHA4RJ0    APL COH2      SATISFACTORY UNAVAILABLE  SOFTDOWN     IDLE        YES      N/A      YES NO
SAPHA4RJ0    APL COH3      SATISFACTORY UNAVAILABLE  SOFTDOWN     IDLE        YES      N/A      YES NO
SAPHA4RJ1    APL COH1      SATISFACTORY UNAVAILABLE  SOFTDOWN     IDLE        YES      N/A      YES NO
SAPHA4RJ1    APL COH2      SATISFACTORY UNAVAILABLE  SOFTDOWN     IDLE        YES      N/A      YES NO
SAPHA4RJ1    APL COH3      SATISFACTORY UNAVAILABLE  SOFTDOWN     IDLE        YES      N/A      YES NO
SAPHA4RM0_X  APG          SATISFACTORY UNAVAILABLE  SOFTDOWN     MOVE        INTERNAL   YES      N/A      YES NO
SAPHA4RM1_X  APG          SATISFACTORY UNAVAILABLE  SOFTDOWN     MOVE        INTERNAL   YES      N/A      YES NO
SAPHA4R0     APG COH1      SATISFACTORY UNAVAILABLE  SOFTDOWN     SERVER      INTERNAL   YES      N/A      YES NO
SAPHA4R0     APG COH2      SATISFACTORY UNAVAILABLE  SOFTDOWN     SERVER      INTERNAL   YES      N/A      YES NO
SAPHA4R0     APG COH3      SATISFACTORY UNAVAILABLE  SOFTDOWN     SERVER      INTERNAL   YES      N/A      YES NO
SAPHA4R1     APG COH1      SATISFACTORY UNAVAILABLE  SOFTDOWN     SERVER      INTERNAL   YES      N/A      YES NO
SAPHA4R1     APG COH2      SATISFACTORY UNAVAILABLE  SOFTDOWN     SERVER      INTERNAL   YES      N/A      YES NO
SAPHA4R1     APG COH3      SATISFACTORY UNAVAILABLE  SOFTDOWN     SERVER      INTERNAL   YES      N/A      YES NO

```

| Figure 50. INGLIST panel where all SAP application server resources are stopped

Stop SAP Central Services Java and ABAP and stop DB2

| SA z/OS monitors the SAP central services Java and ABAP as well as DB2. Based on the active SA z/OS policy, it watches all resources and keeps them at the intended operation level.

- | 1. Log on to a NetView console on your z/OS and use the SA z/OS command INGLIST to display a list of all SA z/OS managed resources.
- | 2. Search for the application group SAPHA4_X, which contains the SAP central services and DB2 resources for SAP system HA4.
- | 3. Set a STOP vote for the SAPHA4_X application group. SA z/OS stops the SAP central services Java and ABAP and also shuts down DB2. Wait for all resources to shut down completely.

| Your SAP environment for SID HA4 has completely shut down and is now ready for maintenance and restart.

Starting an SAP system with SA z/OS

| After a successful completion of any required maintenance, you can use the provided instructions to restart your SAP system.

| After following the instructions from “Stopping an SAP system with SA z/OS” on page 243, you can perform the restart sequence in the following order:

- | 1. **Start DB2 and the SAP central services Java and ABAP:**
 - | a. In Tivoli NetView, use the INGVOTE SAPHA4_X command to check the system automation votes against the HA4 resources.
 - | b. Cancel or kill the STOP vote against the SAPHA4_X resource group and check that all HA4 resources on z/OS start successfully.
- | 2. **Start the SAP application servers:**

- a. In Tivoli NetView, use the INGVOTE SAPHA4RAS_X command to display the system automation vote against the SAP application servers resource group.
- b. Cancel or kill the STOP vote against the SAPHA4RAS_X resource group and check that all SAP application server resources start successfully. The INGLIST panel once again displays the content shown in Figure 49 on page 244.

Your SAP system started and is ready for use.

Stopping an SAP system with System Automation for Multiplatforms and SA z/OS

For any required maintenance, you can use the provided instructions to stop your SAP system with the help of SA MP and SA z/OS.

In this scenario, you manage a highly available SAP system with System Automation for Multiplatforms and SA z/OS. You use two products on different platforms to start and stop your SAP system.

The following sections show examples how to use the command line interface of System Automation for Multiplatforms to start and stop application server resources and other utilities running under SA MP control. Examples to how to use SA z/OS commands to start and stop z/OS resources are also provided in the following sections. The sample system used in the following sections is a single-stack (ABAP) SAP system HA1.

To stop an SAP system, you must follow a specific sequence:

- Stop SAP application servers.
- Stop SAP Central Services ABAP.
- Stop DB2.

The following procedure describes how to stop the SAP system in more detail:

Stop SAP application servers

System Automation for Multiplatforms actively maintains the SAP resources in an System Automation for Multiplatforms domain. Based on the active policy it watches all resources and keeps them at the intended operation level, which can be online (started) or offline (stopped)

1. Log on to the operating system with an authorized user to change the status of the System Automation for Multiplatforms managed resources of your SAP system. Use the command `lssam` to find out about online SAP resources. The following screen is displayed:


```

$ lssam
Online IBM.ResourceGroup:SAP_ABAP_HA1_D13_ihlsco2 Nominal=Online
|- Online IBM.ResourceGroup:SAP_ABAP_HA1_D13_ihlsco2_AS Nominal=Online
'-' Online IBM.Application:SAP_ABAP_HA1_D13_ihlsco2_as:ihlsco2
'-' Online IBM.ResourceGroup:SAP_ABAP_HA1_D13_ihlsco2_SRV Nominal=Online
'-' Online IBM.Application:SAP_ABAP_HA1_D13_ihlsco2_sapstartsrv:ihlsco2
Online IBM.ResourceGroup:SAP_ABAP_HA1_D14_ihlsco3 Nominal=Online
|- Online IBM.ResourceGroup:SAP_ABAP_HA1_D14_ihlsco3_AS Nominal=Online
'-' Online IBM.Application:SAP_ABAP_HA1_D14_ihlsco3_as:ihlsco3
'-' Online IBM.ResourceGroup:SAP_ABAP_HA1_D14_ihlsco3_SRV Nominal=Online
'-' Online IBM.Application:SAP_ABAP_HA1_D14_ihlsco3_sapstartsrv:ihlsco3
Online IBM.ResourceGroup:SAP_ABAP_HA1_DVEBMGS12_ihlsco1 Nominal=Online
|- Online IBM.ResourceGroup:SAP_ABAP_HA1_DVEBMGS12_ihlsco1_AS Nominal=Online
'-' Online IBM.Application:SAP_ABAP_HA1_DVEBMGS12_ihlsco1_as:ihlsco1
'-' Online IBM.ResourceGroup:SAP_ABAP_HA1_DVEBMGS12_ihlsco1_SRV Nominal=Online
'-' Online IBM.Application:SAP_ABAP_HA1_DVEBMGS12_ihlsco1_sapstartsrv:ihlsco1
Online IBM.ResourceGroup:SAP_HOST_AGENT_ihlsco1 Nominal=Online
'-' Online IBM.Application:SAP_HOST_AGENT_ihlsco1_ha:ihlsco1
Online IBM.ResourceGroup:SAP_HOST_AGENT_ihlsco2 Nominal=Online
'-' Online IBM.Application:SAP_HOST_AGENT_ihlsco2_ha:ihlsco2
Online IBM.ResourceGroup:SAP_HOST_AGENT_ihlsco3 Nominal=Online
'-' Online IBM.Application:SAP_HOST_AGENT_ihlsco3_ha:ihlsco3

```

2. To stop all SAP application server resources, use the chrg command with a suitable selection string, such as:

```
chrg -o offline -s "Name like 'SAP_ABAP_HA1_D%'"
```
3. Issue the lssam command to verify the result. The following screen is displayed:

```

$ lssam
Offline IBM.ResourceGroup:SAP_ABAP_HA1_D13_ihlsco2 Nominal=Offline
|- Offline IBM.ResourceGroup:SAP_ABAP_HA1_D13_ihlsco2_AS Nominal=Offline
'-' Offline IBM.Application:SAP_ABAP_HA1_D13_ihlsco2_as:ihlsco2
'-' Offline IBM.ResourceGroup:SAP_ABAP_HA1_D13_ihlsco2_SRV Nominal=Offline
'-' Offline IBM.Application:SAP_ABAP_HA1_D13_ihlsco2_sapstartsrv:ihlsco2
Offline IBM.ResourceGroup:SAP_ABAP_HA1_D14_ihlsco3 Nominal=Offline
|- Offline IBM.ResourceGroup:SAP_ABAP_HA1_D14_ihlsco3_AS Nominal=Offline
'-' Offline IBM.Application:SAP_ABAP_HA1_D14_ihlsco3_as:ihlsco3
'-' Offline IBM.ResourceGroup:SAP_ABAP_HA1_D14_ihlsco3_SRV Nominal=Offline
'-' Offline IBM.Application:SAP_ABAP_HA1_D14_ihlsco3_sapstartsrv:ihlsco3
Offline IBM.ResourceGroup:SAP_ABAP_HA1_DVEBMGS12_ihlsco1 Nominal=Offline
|- Offline IBM.ResourceGroup:SAP_ABAP_HA1_DVEBMGS12_ihlsco1_AS Nominal=Offline
'-' Offline IBM.Application:SAP_ABAP_HA1_DVEBMGS12_ihlsco1_as:ihlsco1
'-' Offline IBM.ResourceGroup:SAP_ABAP_HA1_DVEBMGS12_ihlsco1_SRV Nominal=Offline
'-' Offline IBM.Application:SAP_ABAP_HA1_DVEBMGS12_ihlsco1_sapstartsrv:ihlsco1
Online IBM.ResourceGroup:SAP_HOST_AGENT_ihlsco1 Nominal=Online
'-' Online IBM.Application:SAP_HOST_AGENT_ihlsco1_ha:ihlsco1
Online IBM.ResourceGroup:SAP_HOST_AGENT_ihlsco2 Nominal=Online
'-' Online IBM.Application:SAP_HOST_AGENT_ihlsco2_ha:ihlsco2
Online IBM.ResourceGroup:SAP_HOST_AGENT_ihlsco3 Nominal=Online
'-' Online IBM.Application:SAP_HOST_AGENT_ihlsco3_ha:ihlsco3

```

Stop SAP Central Services ABAP and stop DB2

Use the same process as described in the list with steps 1 on page 245 through 3 on page 245.

Starting an SAP system with System Automation for Multiplatforms and SA z/OS

After a successful completion of any required maintenance, you can use the provided instructions to restart your SAP system with the help of SA MP and SA z/OS.

The restart sequence is performed in the following order:

1. Start DB2 and the SAP Central Services.
2. Start the SAP application servers.

The following steps describe the procedure in detail:

1. **Start DB2 and the SAP Central Services Java and ABAP.**

- a. In Tivoli NetView use the INGVOTE command to check the system automation votes against the HA1 resources.
- b. Cancel or kill the STOP vote against the SAPHA1_X resource group and check that all HA1 resources start again.

2. **Start SAP application servers.**

- a. Log on to the operating system with an authorized user to change the status of the System Automation for Multiplatforms managed resources of your SAP system.
- b. Use the command lssam to verify the status of the SAP resources. The same screen as shown in the last step of this procedure is displayed.
- c. Start all SAP application server resources with the same chrg command you used to stop them but now use the online option:

```
chrg -o online -s "Name like 'SAP_ABAP_HA1_D%'"
```

Wait for all application server instances to start up completely.

- d. Issue the lssam command again. Your SAP system is started now and ready for use.

Checking the replication status

Read this additional information to learn how to determine whether the enqueue table is completely replicated at startup and during runtime.

When using standard replication with a separate ERS, the replication status is initially checked at startup of the ERS resource. The start_cs script uses the SAP utility ensmon to check if the replication is current. Only when this is the case, it returns with return code 0 and issues the message *ERS<xx> instance now ACTIVE*. This causes the status of the SA z/OS resource to become AVAILABLE.

When using *EnqCF replication* (see Chapter 13. Enqueue replication into a z Systems coupling facility), then the replication status is initially checked at startup by the ENQ server itself. The enqueue server terminates itself if it is not able to do an initial replication into the CF.

To make sure that replication is active at runtime, you should use the SAP<SID>AEN_M monitor resource. It requires SAP Kernels 7.40 and higher. For more details see "Monitoring the health state of SAP enqueue replication" on page 166.

If you want to manually determine whether the enqueue table is completely replicated, use the ensmon utility command to determine the status of the replication. A return code of 0 indicates that the replication is active and current. Return code 4 means it is not current.

Change management during SAP operation

This information is about planned maintenance tasks for the hardware or software components that are necessary for your SAP operation. In an ideal scenario such maintenance activities have little or no impact on the running SAP system and its users. The subsections in this topic provide information on how to apply nearly non-disruptive change management within your SAP system for a variety of possible operations.

The procedures in this section describe several common SAP change management operations:

- How to do maintenance with minimum impact
- How to avoid interference of maintenance actions with system automation policies
- How to maintain the highest possible level of availability even during maintenance operations

The following topics are described in detail:

- “Updating DB2 or z/OS”
- “Change management on z/OS UNIX” on page 264
- “Change management on SAP application servers” on page 265
- “SAP tools and restarting SAP application server instances” on page 268
- “Maintenance that requires a reboot of an SAP application server” on page 268
- “SAP Software Update Manager (SUM)” on page 270
- “Updating the SAP kernel” on page 271
- “Updating the SAP kernel using a disruptive procedure” on page 272
- “Updating the SAP kernel using the rolling kernel switch” on page 272
- “Running the rolling kernel switch while automation is active” on page 274

You can also read the following article for further information about connections to a DB2 database and about the setup of a failover configuration environment: *Setup of New DB2 Data-Sharing Failover Solution with the SAP Failover Configuration Tool*.

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. The topic provides information on how to upgrade DB2 and z/OS without system interrupts.

SMP/E is the system tool that is 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 running from different executable libraries. To activate the latest changes, the software (z/OS or DB2) must be stopped and restarted by using the updated loadlibs. For more details about how to apply software maintenance that uses 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. This means you can run multiple software releases in a Parallel Sysplex data sharing environment. z/OS normally supports N-2 releases. This means that up to three consecutive releases can run in the same Parallel Sysplex.

DB2 allows you to non-disruptively upgrade to a higher release. For example, DB2 11 for z/OS can run in parallel with DB2 12 in the same data sharing group as long as the new function (V12R1M500) of DB2 12 is not activated. By iteratively moving all DB2 members of the data sharing group to DB2 12, and staying at function level V12R1M100, the SAP application can remain online all the time. When all of the DB2 members of the data sharing group have been moved, you can execute the **-ACTIVATE FUNCTION LEVEL** command for V12R1M500. All of these activities can be run without stopping SAP.

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 both being used, individual z/OS LPARs and DB2 data sharing members can be started and stopped without stopping the SAP system. This is accomplished by taking advantage of DB2 connection failover, see “DB2 connection failover” on page 15. Complete the following steps for each LPAR in the sysplex to be updated:

1. **Build new DB2 loadlibs** with the DB2 maintenance that is applied for each DB2 data sharing member. A suggested name would be :
`<db2_member_name>.SDSNLOAD.NEW`
2. **Stop SAP batch processing.** Make sure that all SAP batch processing on application servers that are connected to DB2 in this LPAR has stopped and no new batch workload gets scheduled. You can use SAP transaction RZ03 to switch these application servers to a previously defined operation mode that does not comprise any SAP batch work processes. Such an operation mode prevents new batch work from getting scheduled on these application servers.
3. **Initiate DB2 connection failover**
 - **Method 1:** For both ABAP and Java instances use the *graceful stop* of the DDF address space - **STOP DDF MODE(QUIESCE)** - to move any database connections of these instances away from the DB2 data sharing member that is going to be updated.
 - **Method 2:** For ABAP instances you can use SAP GUI transaction DBACOCKPIT to move the instances away from the DB2 data sharing member that is going to be updated. The SAP report RSDB2SWITCH or the RFC utility **rftcmove** can also be used to achieve this result. The alternatives of using RSDB2SWITCH or **rftcmove** are described in more detail in “Switching database connections for multiple SAP ABAP application servers” on page 261.
4. **Stop the DB2 data sharing members in the LPAR.** Ensure that there are no active connections to this DB2 member before issuing the **Stop DB2** command. Alternatively, you can use SA z/OS to stop the DB2 data sharing members.
5. **Switch from current DB2 loadlibs to new DB2 loadlibs.** This can, for example, be accomplished by renaming the DB2 load libraries.
6. **Stop and re-IPL z/OS to activate z/OS updates.** At this point, z/OS can be stopped and re-IPLed to activate z/OS updates. Other services that were running on this LPAR, such as the NFS server, SAP Web Dispatcher, SAP Router, SAP Central Services, should be restarted automatically on one of the other active LPARs by System Automation. This assumes that your automation policy works correctly and has been verified (see Chapter 10, “Verifying your implementation on z/OS,” on page 193).
7. **Restart the DB2 data sharing members in the LPAR.**
8. **Switch back ABAP application servers to the previous configuration by using SAP transaction DBACOCKPIT.**

- **Method 1:** Database connections of both ABAP or Java instances automatically switch back to the restarted DB2 member, if **affinityFailbackInterval** parameter was set in the connection profile.
 - **Method 2:** Switch back ABAP application servers using SAP transaction DBACOCKPIT, SAP report RSDB2SWITCH or the RFC utility **rfcmove**.
9. Verify the switched back threads to the restarted DB2 member.
 10. **Restart all SAP batch processing** on application servers that are connected to DB2 in this LPAR. Use Opt Mode Switch to add batch work processes.

Switching database connections for a single SAP application server

In this example the SAP DB2 database is set up as a data sharing system with three members:

- SE51
- SE52
- SE53

There are also three SAP application servers, and each application server has a primary and a secondary connection to a dedicated DB2 member:

SAP AS ServerName	SAP AS hostname	DB2 member
ih1scoh1v_HA1_12	ih1scoh1v	primary database connection to SE51 alternate database connections to SE52, SE53
ih1scoh2v_HA1_13	ih1scoh2v	primary database connection to SE52 alternate database connections to SE53, SE51
ih1scoh3v_HA1_14	ih1scoh3v	primary database connection to SE53 alternate database connections to SE51, SE52

The primary connection is the initial connection. The alternate connections are used for the failover.

The following example is based on the assumption that all SAP application servers are connected to their primary DB2 member, in particular ih1scoh1v is connected to SE51.

The goal of this example is to apply maintenance for DB2 member SE51 in such a way that there is no impact for SAP users that are logged on to SAP application server ih1scoh1v.

Assume that step **1. Build new DB2 loadlibs** (build new DB2 loadlibs, or any other DB2 and/or z/OS preparations) and step **2. Make sure that all SAP batch processing on application servers that are connected to DB2 in this LPAR has stopped and no new batch workload gets scheduled.** from topic "Updating DB2 or z/OS" on page 249 have already been completed.

Prerequisites for planned DB2 failover of application server ih1scoh1v

A connection profile that contains failover connections must be defined and activated as a prerequisite before step **3. Activate DB2 connection failover** from topic "Updating DB2 or z/OS" on page 249.

Defining connection profiles

SAP Systems installed with SWPM 1.0 SP18 or newer versions have a connection profile for the ABAP stack, called **db2dsdriver.cfg** (older SAP ABAP Systems may base on SAP failover and its connection profile `connect.ini`). For the Java stack, the connection profile is called `config.xml`. Each connection profile contains the possible connections between the SAP application servers and the DB2 members. The SAP application server builds up its DB connection in the same sequence as is listed in the connection profile.

For more information and references about how to configure the **CLI failover**, see “CLI failover for ABAP instances” on page 16.

In this example, the ABAP connection profile **db2dsdriver.cfg** displays three possible connections to the database for each SAP application server. If the primary DB2 member is not reachable, the application server attempts to establish the connection with the second member. If this second member also cannot be reached, it attempts to establish the connection with the third DB2 member.

Example **db2dsdriver.cfg** file:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <dsncollection>
    <dsn alias="HA1" name="SAPHA1" host="coh1vipa.boeblingen.de.ibm.com" port="1111"/>
  </dsncollection>
  <databases>
    <database name="SAPHA1" host="coh1vipa.boeblingen.de.ibm.com" port="1111">
      <acr>
        <parameter name="Authentication" value="SERVER" />
        <parameter name="acrRetryInterval" value="0"/>
        <parameter name="affinityFailbackInterval" value="300"/>
        <parameter name="enableAcr" value="true"/>
        <parameter name="enableSeamlessAcr" value="true"/>
        <parameter name="maxAcrRetries" value="3"/>
        <alternateserverlist>
          <server name="SE51" hostname="coh1vipa.boeblingen.de.ibm.com" port="1111"/>
          <server name="SE52" hostname="coh2vipa.boeblingen.de.ibm.com" port="1111"/>
          <server name="SE53" hostname="coh3vipa.boeblingen.de.ibm.com" port="1111"/>
        </alternateserverlist>
        <affinitylist>
          <list name="list1" serverorder="SE51,SE52,SE53"/>
          <list name="list2" serverorder="SE52,SE53,SE51"/>
          <list name="list3" serverorder="SE53,SE51,SE52"/>
        </affinitylist>
        <clientaffinitydefined>
          <client name="APP1" hostname="ihlscohlv" listname="list1"/>
          <client name="APP2" hostname="ihlscoh2v" listname="list2"/>
          <client name="APP3" hostname="ihlscoh3v" listname="list3"/>
        </clientaffinitydefined>
      </acr>
    </database>
  </databases>
</configuration>
```

The **db2dsdriver.cfg** file contains sections where you define your data source, the database connection, and the client information. From the CLI point of view, the SAP application server hosts are the clients.

Within the `<dsncollection>` section, you define the data source name (`<dsn>`) with an arbitrary alias. The `<database>` section contains definitions of the database location. In the `<acr>` (*automatic client reroute*) subsection, you define the parameters, which influence the **CLI failover** behavior.

In the example, the SAP system identifier HA1 was chosen as an alias for the data source name.

In the <database> section, the database location with the DB2 dynamic location alias SAPHA1 on host **coh1vipa**, is available via port 1111. The DDF port must be identical for all members of the data sharing group.

In the <parameter> section, the **affinityFailbackInterval** parameter is set to 300 seconds. This is a valid setting if you trigger the planned DB2 failover from the DB2 server side via the command **STOP DDF** (see **Method 1** from Step 3 on page 250: **Initiate DB2 connection failover**). The setting has the effect that once the DB2 server has been restarted, the SAP work processes automatically fails back to their primary member within this interval. To control the DB2 planned failovers via SAP DBACOCKPIT or ABAP report RSDB2SWITCH, the parameter **affinityFailbackInterval** needs to be set to 0.

Note: SAP installation for ABAP instances inserts a value of 0 as a default value for this parameter into the connection profile.

In the <alternateserverlist> subsection, the DB2 members SE51, SE52, and SE53 are defined with their virtual host names.

In the <affinitylist> section, three connection lists are defined. Each list defines a parameter **serverorder** which is an ordered list of DB2 members. In case of a connection failure, the DB2 CLI driver reroutes the client database connections according to this order.

In the <clientaffinitydefined> section, you assign a list with its **serverorder** to an application server host.

For example, the DB2 CLI driver on host **ih1scoh1v** uses the routing that is defined by **list1**, which means that it primarily connects to the DB2 member SE51. If SE51 is not accessible, it attempts to establish the connection with SE52. If SE52 also is not accessible, it connects to SE53.

The same principle applies for the application servers that use APP2 and APP3, with the only difference, that they have another routing list, each with another DB2 member sequence.

Example: DDF definition on host coh1vipa for DB2 member SE51 which has a dynamic alias SAPHA1 defined with port 1111:

```

-SE51 DSNLTDDF DISPLAY DDF REPORT FOLLOWS:
STATUS=STARTD
LOCATION          LUNAME          GENERICLU
COHDBE5         DEIBMIPS.IPXAOE51 -NONE
TCPSPORT=11050 SECPOR=0        RESPORT=11056 IPNAME=-NONE
IPADDR=:10.101.4.210
SQL             DOMAIN=coh1vipa.boeblingen.de.ibm.com
RESYNC DOMAIN=coh1vipa.boeblingen.de.ibm.com
ALIAS          PORT  SECPOR  STATUS
COHDBE5SE51   11051 0      STATIC
SAPHA1       1111 0      STARTD
MEMBER IPADDR=:10.101.4.210
DT=I  CONDBAT= 10000 MDBAT= 200
MCONQN= 0 MCONQW= 0
ADBAT= 17 QUEDBAT= 0 INADBAT= 0 CONQUED= 0
DSCDBAT= 0 INACONN= 4
WLMHEALTH=100 CLSDCONQN= 0 CLSDCONQW= 0
LOCATION SERVER LIST:
WT IPADDR      IPADDR
51 :10.101.4.211
...
CURRENT DDF OPTIONS ARE:
PKGREL = COMMIT
DSNLTDDF DISPLAY DDF REPORT COMPLETE

```

Theoretically, you can define one `db2dsdriver.cfg` file, or multiple files with different names, for example, one per SAP application server. It is recommended that you use only one `db2dsdriver.cfg` file and place it into the SAP directory `/sapmnt/<sid>/global`. This has the advantage that you must maintain only one connection configuration file. In addition, the SAP global file system can always be accessed because it is exported by the highly available NFS Server on z/OS.

For the Java stack, the connection profile `config.xml` looks similar to the `db2dsdriver.cfg` file and has the same effect. The `config.xml` file is generated by the SAP installation and written into the directory `/sapmnt/<sid>/global`.

For information about how to modify the `db2dsdriver.cfg` or the `config.xml` file, see the *Database Administration Guide: SAP on IBM DB2 for z/OS*.

Verification activities before the switch

Before you remove the DB2 connections from DB2 member SE51 as described in step 3. **Activate DB2 connection failover** from topic “Updating DB2 or z/OS” on page 249, you need to verify the current SAP application server distribution.

You can do this with the MVS command `DISPLAY THREAD` for each DB2 member. Here is an example for DB2 member SE51:


```

DSNV401I -SE51 DISPLAY THREAD REPORT FOLLOWS -
DSNV402I -SE51 ACTIVE THREADS -
NAME      ST A  REQ ID          AUTHID  PLAN    ASID  TOKEN
SERVER    RA * 162 HA1DIA000    R3USER  DISTSERV 00ED 26612
V437-WORKSTATION=*
        USERID=SAPSYS
        APPLICATION NAME=*
V442-CRTKN=10.101.4.214.15176.140915084305
V445-GA6504D6.JB48.140915084305=26612 ACCESSING DATA FOR
        ::FFFF:10.101.4.214
SERVER    RA * 78 HA1DIA001    R3USER  DISTSERV 00ED 26610
V437-WORKSTATION=*
        USERID=SAPSYS
        APPLICATION NAME=*
V442-CRTKN=10.101.4.214.15090.140915084102
V445-GA6504D6.JAF2.140915084102=26610 ACCESSING DATA FOR
        ::FFFF:10.101.4.214

```

The command output displays useful information about each thread, for both Java threads and ABAP threads.

For example:

- The thread ID=HA1DIA000 shows some characteristics of the ABAP work process:
 - HA1 is the SAP system identifier
 - DIA indicates that the type of the SAP work process is an SAP dialog work process
 - 000 indicates the SAP work process number (see SAP transaction SM50). It also indicates the number of the SAP developer trace file (dev_w00)
- The AUTHID=R3USER (DB2 schema for ABAP stack) states that this is a thread connection from ABAP stack. In case the thread comes from a Java stack. The AUTHID would be the DB2 schema SAPJAVA.
- The V442-CRTKN information shows the virtual IP address (10.101.4.214) of the SAP application server host ih1scoh1v.

The DISPLAY THREAD command output for DB2 members SE52 and SE53 shows similar connections from the application server hosts ih1scoh2v and ih1scoh3v.

You can also check the current database connections of the SAP ABAP application server instances in your SAPGUI using SAP transaction DBACOCKPIT. Navigate to **DB Connections** where the following information is displayed:

- **DB Connection List:** the list shows all possible DB connections from the SAP application server you are currently logged on. You must switch DB connections to another DB2 member from here.
- **Active DB Connections:** shows current DB connections for this SAP application server.
- **Data Sharing Topology:** shows the current correlation of all SAP application servers and DB2 members

In **DBA Cockpit: System Configuration Maintenance**, open **Diagnostics** and **DB connections**, double-click either DB Connection List or Active DB Connections or Data Sharing Topology.

Note: The following example refers only to the SAP transaction DBACOCKPIT.

For example, the DB2 connection list for SAP application server on host ihlscoh1v (see Figure 51) shows only the possible database connections, which are defined for this application server in the `db2dsdriver.cfg` file.

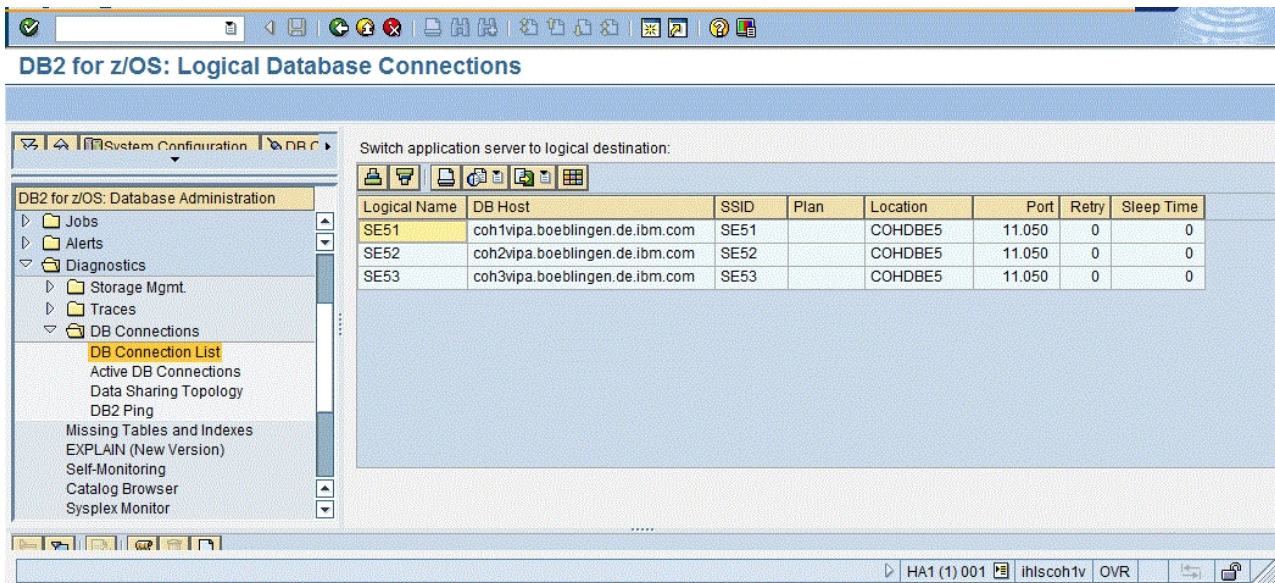


Figure 51. DB connection list for the SAP application server on host ihlscoh1v

The logical connection is a symbolic name for one possible DB connection and contains all the relevant information for this connection. In Figure 51, the **Logical Name** SE51 corresponds to the server name SE51 in the sample `db2dsdriver.cfg` file on page 252.

The *Active DB Connections* panel for SAP application server on host ihlscoh1v shows the connection of work process:

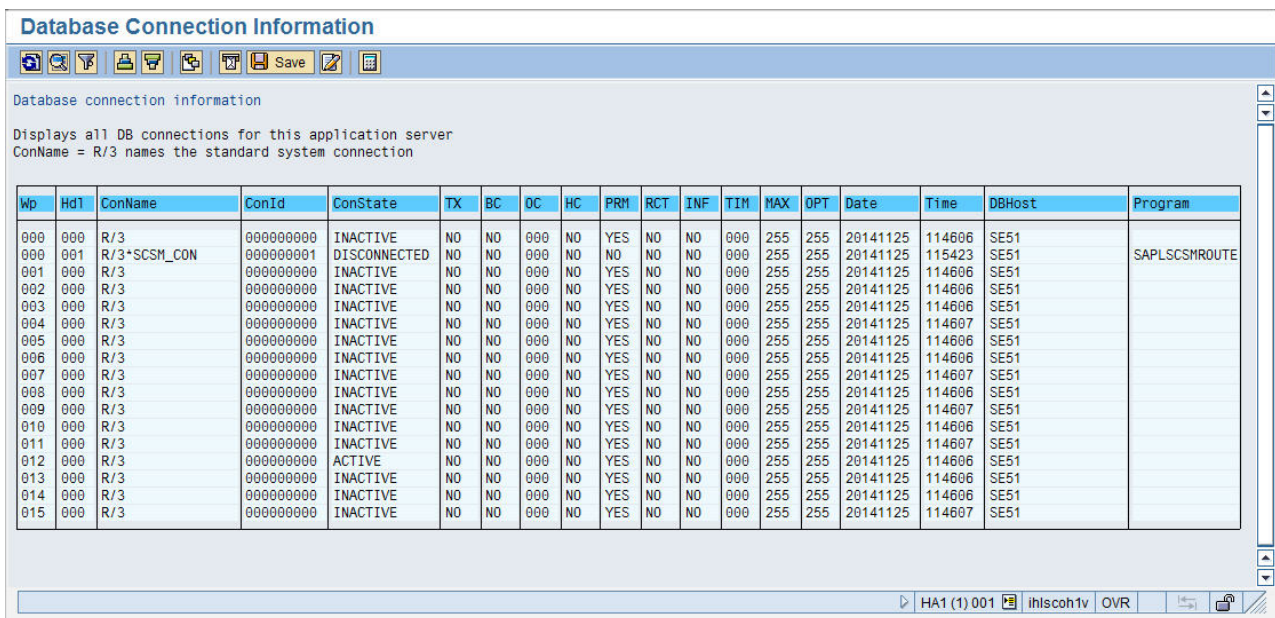


Figure 52. Active DB connections

Only the ACTIVE and INACTIVE connections are relevant. Connections with status DISCONNECTED are just information from the memory buffer.

The **Data Sharing Topology** shows the current database connection that each SAP ABAP application server is using:

The screenshot shows the 'DB2: Data Sharing Topology-Correlation AppServer/DB2 Member' window. The left pane shows the navigation tree with 'Data Sharing Topology' selected. The main pane displays a table with the following data:

DB2 member/SAP app.server/Work process	Dialog	Update	Update2	Background	Enqueue	Spool	Java
SE51(SSID:SE51, z/OS:COH1)	10	1	1	3	0	1	0
ihlsco1v_HA1_12	10	1	1	3	0	1	0
Work process 0	1	0	0	0	0	0	0
Work process 1	1	0	0	0	0	0	0
Work process 2	1	0	0	0	0	0	0
Work process 3	1	0	0	0	0	0	0
Work process 4	1	0	0	0	0	0	0
Work process 5	1	0	0	0	0	0	0
Work process 6	1	0	0	0	0	0	0
Work process 7	1	0	0	0	0	0	0
Work process 8	1	0	0	0	0	0	0
Work process 9	1	0	0	0	0	0	0
Work process 10	0	1	0	0	0	0	0
Work process 11	0	0	0	1	0	0	0
Work process 12	0	0	0	1	0	0	0
Work process 13	0	0	0	1	0	0	0
Work process 14	0	0	0	0	0	1	0
Work process 15	0	0	1	0	0	0	0
SE52(SSID:SE52, z/OS:COH2)	10	1	1	3	0	1	0
ihlsco2v_HA1_13	10	1	1	3	0	1	0
SE53(SSID:SE53, z/OS:COH3)	10	1	1	3	0	1	0
ihlsco3v_HA1_14	10	1	1	3	0	1	0

Figure 53. Data sharing topology

The analysis of DB connections shows that only the ABAP work processes from the SAP application server host ihlsco1v are connected to DB2 member SE51. Before DB member SE51 can be stopped for maintenance, the connections and DB2 threads must be moved from SE51 to another DB member.

Switch using SAP transaction DBACOCKPIT

The subsequent information describes how to initiate a database connection switch using SAP transaction DBACOCKPIT.

Moving ABAP application server threads from SE51 using DBACOCKPIT

To perform the connection move, you need to be logged-on in SAPGUI to that SAP application server, from which you want to switch away the DB connection.

Note: Ensure to set the parameter `affinityFallbackInterval` to 0 before you use DBACOCKPIT to move database connections.

From the **DB2 Subsystem Activity** screen, select **Diagnostics** and **DB connections**, then double-click the **DB Connection List**.

To move the threads from DB2 member SE51 to target member SE52, select the target database connection with logical name SE52:

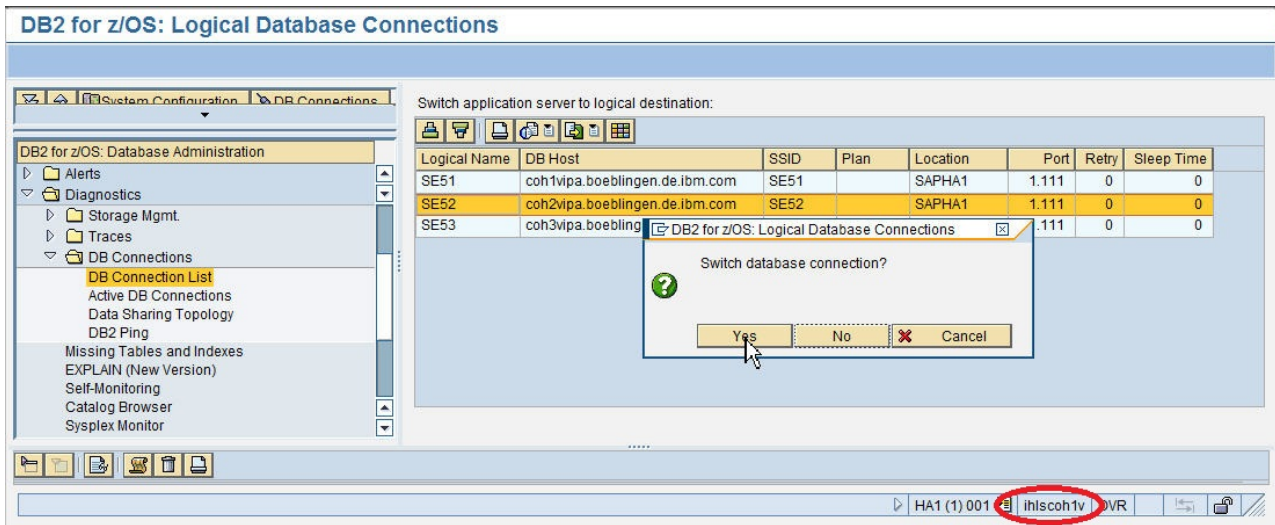


Figure 54. Select the target connection name and switch database connection

Click **Yes** to confirm the connection switch. The SAP information is displayed that the database failover to target DB member was issued.

The SAP connections from SAP application server host ihlsco1v moves from DB2 member SE51 to SE52.

Note: Only the DB connection from the currently logged on SAP application server is switched. The DB connections for the other application servers remain unchanged.

You can verify this for ABAP work processes in the SAP transaction DBACOCKPIT and from the information in **Data Sharing Topology**:

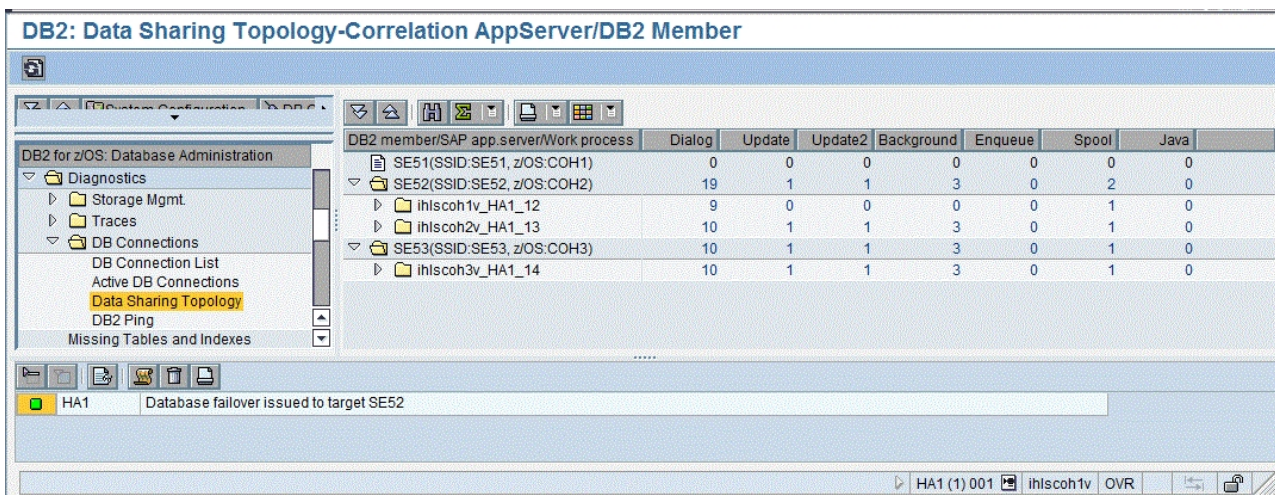


Figure 55. ABAP work processes information in the Data Sharing Topology view

Because no DB2 threads are connected to DB2 subsystem SE51, the subsystem can now be stopped with system automation, see step 4 (stop the DB2 data sharing members in the LPAR).

Switch using SAP report RSDB2SWITCH

To move away the ABAP database connection, you can alternatively use the SAP report RSDB2SWITCH.

Note: Ensure to set the parameter **affinityFailbackInterval** to 0 before you use RSDB2SWITCH to move database connections.

Start the report from transaction SE38 and enter the parameters:

NEWDBCON

Enter the logical name of the target database. This entry is case sensitive and therefore must be entered in the same way as the server name is defined in the **db2dsdriver.cfg** file.

ALLAPPS

leave this entry field blank if only the DB connection of the current logged on SAP application server should be moved to the new DB connection (NEWDBCON). If you enter X, then the database connections of all SAP application servers move to the new DB connection.

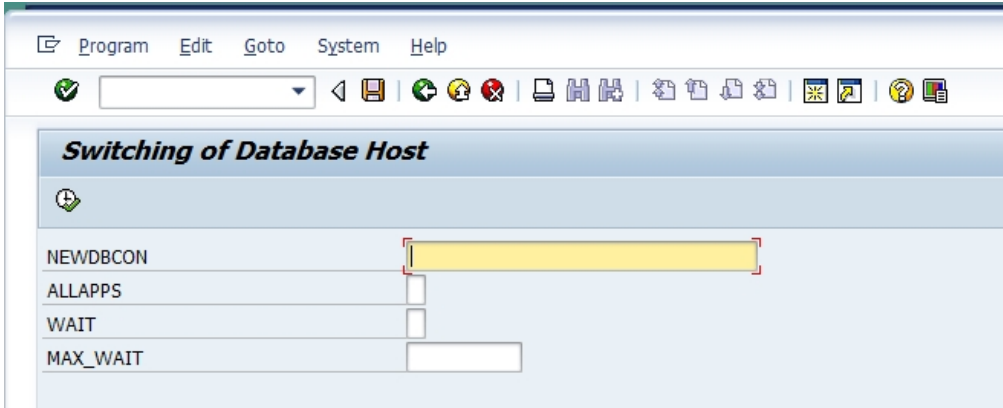
WAIT When you enter X, then the report provides feedback only when all DB connections have been moved to the new DB connection.

MAX_WAIT

Enter the time (in seconds) after which the report should end and show the status of moved DB connections. If not all connections have been moved restart the report until all DB connections have been moved.

The parameters WAIT or MAX_WAIT make sense especially when the report is scheduled as a batch job.

Example of RSDB2SWITCH and its entry fields:



The screenshot shows the SAP report RSDB2SWITCH parameter entry screen. The window title is "Switching of Database Host". The screen displays four input fields: NEWDBCON, ALLAPPS, WAIT, and MAX_WAIT. The NEWDBCON field is highlighted with a yellow background, indicating it is the current focus. The ALLAPPS, WAIT, and MAX_WAIT fields are currently empty.

Figure 56. Select the target connection name and switch database connection

Using RSDB2SWITCH has the same effect as switching the database connection via SAP transaction DBACOCKPIT: the connection moves to member SE52 .

Switch using STOP DDF

Database connections can be *gracefully* stopped by issuing the **STOP DDF MODE(QUIESCE)** command. For ABAP instances this is an alternative to switching database connections via DBACOCKPIT or RSDB2SWITCH. For Java instances, **STOP DDF** is the only mechanism you can use to stop database connections and trigger a planned failover.

This example shows the connection move away from DB2 member SE51:

```
-SE51 STOP DDF MODE(QUIESCE)
DSNL021I -SE51 STOP DDF COMMAND ACCEPTED
DSNL005I -SE51 DDF IS STOPPING
DSNL006I -SE51 DDF STOP COMPLETE
```

There are no longer any threads that are connected to DB2 member SE51:

```
-SE51 DIS THREAD(*) LIMIT(*)
DSNV401I -SE51 DISPLAY THREAD REPORT FOLLOWS -
DSNV419I -SE51 NO CONNECTIONS FOUND
DSN9022I -SE51 DSNVDT '-DIS THREAD' NORMAL COMPLETION
```

Stopping a DB2 member

As soon as there are no more connections the DB2 member can be stopped. If your DB2 subsystem is under SA z/OS control, you need to use SA z/OS commands to stop a member. For the SAP System HA1, the first DB2 member SE51 is modelled in SA z/OS application group HA11_X/APG. In order to stop member SE51, issue a STOP vote for this group. As soon as HA11_X/APG is unavailable (stopped), consider changing its SA z/OS AUTOMATION FLAG to NO. Automation then does not take any action if the DB2 member has to be restarted manually in maintenance mode which may be required as part of the DB2 maintenance.

Switching back database connections

First, you must restart the DB2 member SE51 either by an explicit operator command, or by resetting the AUTOMATION FLAG and canceling the *STOP vote* for HA11_X/APG in SA z/OS.

The DB2 member restarts.

If you used **STOP DDF** to stop Java or ABAP database connections, then a restart of the DB2 member triggers a reconnect of Java or ABAP connections to this member, provided the parameter **affinityFailbackInterval** was set to a positive value in the config.xml (Java) or db2dsdriver.cfg (ABAP) configuration files.

ABAP connections of ih1scoh1v can be explicitly switched back with SAP transaction DBACOCKPIT as follows:

- Log on to SAP application server on host ih1scoh1v. From the SAP transaction DBACOCKPIT open **Diagnostics** and **DB connections**, then double-click the **DB Connection List**. Double-click the target DB connection, which is the DB2 member SE51 with target server name SE51 as defined in the db2dsdriver.cfg file (see the db2dsdriver.cfg example on page 252).

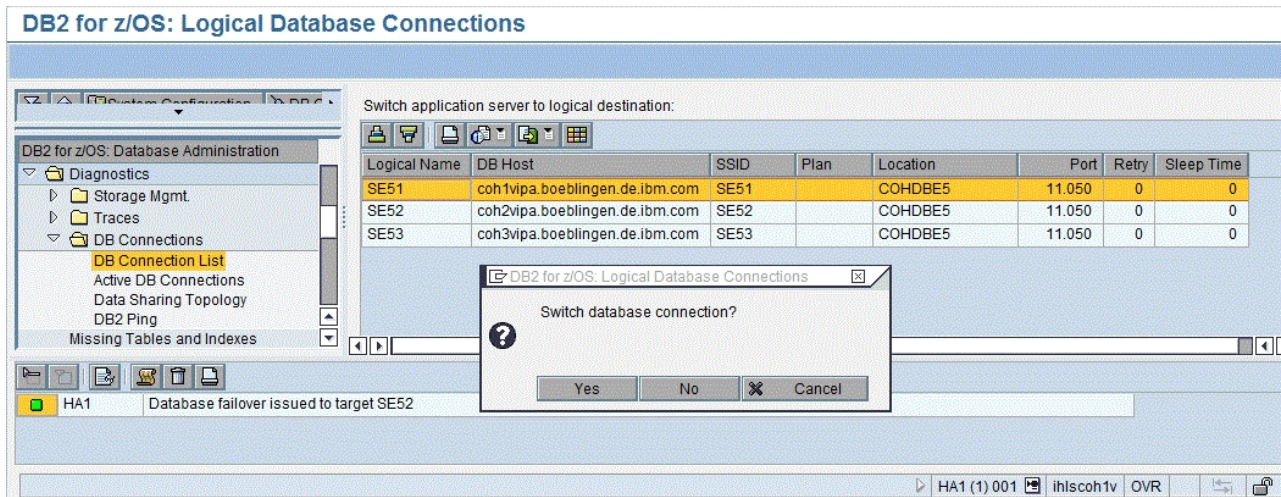


Figure 57. Double-click the target DB connection

- Alternatively, you can use the SAP report RSDB2SWITCH to switch back the DB connections. Log on to the SAP application server for which the DB connections must be moved back, enter the primary DB connection, and run the report.

Verify that the ABAP threads from the application server host ih1scoh1v are switched back to its primary DB member SE51 (step 9 from the procedure in topic “Updating DB2 or z/OS” on page 249). To do this use the DB2COCKPIT or DIS THREAD command as previously described.

Restart the SAP batch processing for SAP application server on host ih1scoh1v SAP transaction RZ03 (step 10 **Restart all SAP batch processing** from topic “Updating DB2 or z/OS” on page 249).

This is the procedure for DB2 member SE51. Repeat the procedure (steps 1 - 10) in topic “Updating DB2 or z/OS” on page 249 for the maintenance of DB2 members SE52 and SE53.

Switching database connections for multiple SAP ABAP application servers

The mechanism that is outlined in “Switching database connections for a single SAP application server” on page 251 describes how to switch the database connection of individual SAP ABAP application servers via SAPGUI.

The following mechanisms might be more suitable for large numbers of SAP ABAP application servers that need to be switched back and forth during DB2 maintenance between primary and secondary DB2 members.

The following examples show a symmetric setup where the SAP DB2 database is set up as a data sharing system with two members, SE51 and SE52. Maintenance is to be performed for member SE51.

In the examples, a total of 10 SAP application servers are initially evenly distributed across these two members. Five application servers have primary connections to SE51 and the other five application servers initially connect to SE52. Secondary connections are defined to the other DB2 member as shown in the samples of failover configuration files.

The assumption in the following example is that ABAP application servers ihlscohlv to ihlscoh5v use the DB2 member SE51 as the primary connection, and the SE52 as secondary connection. For the application servers ihlscoh6v to ihlscoh10v, it is the other way around. This means that their primary database connection goes to DB2 member SE52. If this member is unavailable, the application servers connect to SE51. Before you start maintenance for DB2 member SE51, all application servers with a primary connection to member SE51 should switch their database connection to SE52.

Sample db2dsdriver.cfg file with a symmetric setup:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
<dsncollection>
<dsn alias="HA1" name="SAPHA1" host="coh1vipa.boeblingen.de.ibm.com" port="1111"/>
</dsncollection>
<databases>
  <database name="SAPHA1" host="coh1vipa.boeblingen.de.ibm.com" port="1111">
    <acr>
      <parameter name="Authentication" value="SERVER" />
      <parameter name="acrRetryInterval" value="0"/>
      <parameter name="affinityFailbackInterval" value="300"/>
      <parameter name="enableAcr" value="true"/>
      <parameter name="enableSeamlessAcr" value="true"/>
      <parameter name="maxAcrRetries" value="3"/>
      <alternateserverlist>
        <server name="SE51" hostname="coh1vipa.boeblingen.de.ibm.com" port="1111"/>
        <server name="SE52" hostname="coh2vipa.boeblingen.de.ibm.com" port="1111"/>
      </alternateserverlist>
      <affinitylist>
        <list name="list1" serverorder="SE51,SE52"/>
        <list name="list2" serverorder="SE52,SE51"/>
      </affinitylist>
      <clientaffinitydefined>
        <client name="APP1" hostname="ihlscohlv" listname="list1"/>
        <client name="APP2" hostname="ihlscoh2v" listname="list1"/>
        <client name="APP3" hostname="ihlscoh3v" listname="list1"/>
        <client name="APP4" hostname="ihlscoh4v" listname="list1"/>
        <client name="APP5" hostname="ihlscoh5v" listname="list1"/>
        <client name="APP6" hostname="ihlscoh6v" listname="list2"/>
        <client name="APP7" hostname="ihlscoh7v" listname="list2"/>
        <client name="APP8" hostname="ihlscoh8v" listname="list2"/>
        <client name="APP9" hostname="ihlscoh9v" listname="list2"/>
        <client name="APP10" hostname="ihlscoh10v" listname="list2"/>
      </clientaffinitydefined>
    </acr>
  </database>
</databases>
</configuration>
```

Each application server host (client) is assigned to an *affinity* contains DB2 members in a certain order. In the example, the clients APP1 to APP5 are assigned to list1, which means, that their primary connection goes to DB2 member SE51. If this is unavailable, they connect to DB2 member SE52. Clients APP6 to APP10 are assigned to list2 where the primary connection goes to DB2 SE52. If SE52 is unavailable, they connect to DB2 SE51.

The DB2 maintenance should be processed with the following steps:

1. Stop DDF of SE51 with MODE(QUIESCE). Thus, the clients APP1 to APP5 with affinity list1 move their database connection to SE52 (according to the member sequence of list1).

2. As soon as all connections have moved to SE52, you can start with the DB2 maintenance on SE51. As long as DDF remains stopped, any new or restarted connections from clients APP1 to APP5 goes to DB2 member SE52.
3. After the DB2 maintenance, start DDF of SE51.
4. As soon as SE51 is available again, the setting of the **affinityFailbackInterval** parameter to a positive number of seconds will trigger a fail back of database connections from APP1 to APP5 back to its primary member SE51 within this time-frame. Alternatively, SAP transaction DBACOCKPIT can be used to explicitly move these connections back.

Note: For SAP Java instances with connection profile `config.xml` the same STOP DDF procedure as for ABAP instances can be used to initiate a mass move.

Mass move for ABAP instances using an ABAP report

You can execute the SAP report RSDB2SWITCH to switch the DB connections to DB2 member SE52 for all ABAP instances. Enter the target database connection SE52 in entry field NEWDBCON and an x for ALLAPPS (to move the connections of all SAP application servers to the new database connection).

As a result, all application servers that have connections to SE51 are moved to DB2 member SE52.

- For application servers APP1 to APP5, this means that all connections switch to their secondary DB2 member SE52.
- For application servers APP6 to APP10, this means no change, because its application servers are already connected to (their primary) DB2 member SE52.

Mass move for ABAP instances using a RFC utility

You can use the RFC utility **rfcmove** to perform such mass move from a script or a program. The sample source code for the **rfcmove** utility is delivered by SAP. You need to compile it once for your environment. Download the file `sample_c_call.txt` attached to *SAP Note 915482:DB2 z/OS: Automating DB failover* and follow the instructions therein.

Rename the file to a meaningful name and compile it with the C-compiler for your platform. In this example `sample_c_callv1.txt` is renamed to `rfcmove.c` and is compiled for z/OS UNIX System Services (z/OS UNIX). When invoked with the `-?` option, **rfcmove** displays a short help text. The following sample command switches all SAP ABAP application servers to DB2 member SE52:

```
rfcmove -h ih1scoh1 -n 02 -u USERIDXX -p xxxxxxxx -c 001 SE52
```

With the password specified after the `-p` option, the supplied SAP user ID USERIDXX logs on to the application server on host `ih1scoh1`, instance number 02. It also logs on to the SAP client 001. The SAP user ID must be authorized in SAP to use RFC functions and call the ABAP function STU3_ADMIN_SWITCH_DB_CON.

As a result, all application servers that have connections to SE51, are moved to DB2 member SE52.

- For application servers APP1 to APP5, this means that all connections switch to their secondary DB2 member SE52.
- For application servers APP6 to APP10, this means no change, because its application servers are already connected to (their primary) DB2 member SE52.

Change management on z/OS UNIX

This information is about update and maintenance tasks for SAP components that run on z/OS UNIX: SAP central services components, SAP profile parameters, and SAP replication servers.

You can perform maintenance tasks for SAP components that run on z/OS UNIX with very limited or no impact to the SAP system. This assumes that your SAP central services can be moved between z/OS LPARs and that enqueue replication is active.

The following scenarios assume that your SAP components on z/OS are controlled by SA z/OS as described in Chapter 9, “Customizing System Automation for Multiplatforms,” on page 185.

The impact that a restart of individual SAP components has on the SAP system, is discussed in detail in “Failure scenarios and impact” on page 83.

Updating SAP central services components

To update one or all of the central services components on z/OS UNIX (the enqueue server or the message server), you must initiate a restart of the central services, which results in a move to the LPAR on which the enqueue replication server is running. Make sure that the restarted components include the new or updated SAP patch level by following this procedure:

1. Save the old modules which reside in the SAP global executable directory and copy the new module(s) to this directory: `/sapmnt/<sid>/exe/uc/<platform>`
2. Trigger a restart of the central services components by moving the central services to the system where the enqueue replication server is running.

The `start_cs` script is responsible for (re)starting central services components. It triggers `sapstart`, which in turn runs first a copy operation from the SAP global executable directory to the instance-specific local directory. `sapstart` uses the standard SAP `sapcpe` mechanism for this purpose. The `sapstart` process then (re)starts the central services components.

With System Automation for z/OS, it is recommended that you issue an `INGREQ STOP RESTART=YES` command against the ABAP or the Java SCS Sysplex Move groups, `SAP<SID>ASCSX` or `SAP<SID>JSCSX`, or both. As a side effect the enqueue replication server moves away from this system and is restarted on another system.

If you are using the new SAP enqueue replication mechanism into the z/OS coupling facility, then you do not require an enqueue replication server, and the enqueue server can be restarted in place on the same LPAR. This requires that you have defined the System Automation policy as described in “Option 2: Using EnqCF replication only” on page 280.

If you need to restart just an ERS instance via System Automation for z/OS, it is recommended that you issue an `INGREQ STOP RESTART=YES` against the ABAP or Java ERS Sysplex Move groups, `SAP<SID>AER_X` or `SAP<SID>JER_X`, or both.

Note: If you did not specify `RESTART=YES` with the `INGREQ`, make sure you remove the `STOP` vote after all components have completed their restart.

Updating SAP profile parameters

To change any of the central services profile parameters, for example, to increase the size of the enqueue table, you should take a similar approach as described in “Updating SAP central services components” on page 264.

1. Modify the central services profile.
2. Restart the central services components as described in step 2. **Make sure that all SAP batch processing on application servers that are connected to DB2 in this LPAR has stopped and no new batch workload gets scheduled.** of topic “Updating DB2 or z/OS” on page 249.

Updating the SAP replication server

If you need to update the SAP replication server, exchange the module as described in “Updating SAP central services components” on page 264.

Issue a subsequent INGREQ STOP with RESTART=YES to make sure that the replication server is restarted on the system on which it was running before.

For the short period during which the replication server is restarted and its SAP enqueue replication table is being rebuilt there is a loss of failover capability until the rebuild is completed. You should therefore schedule such a replacement for a time when there is low enqueue activity in the system. Use SAP's ensmn utility or SAP transaction SM12 to monitor the enqueue activity and the number of SAP enqueue locks. To determine when the restarted replication server has caught up with the enqueue server, see “Checking the replication status” on page 248.

Change management on SAP application servers

The various aspects of maintaining SAP application servers are discussed in the sections of this topic. They comprise operating system or hardware maintenance and updates of the DB2 CLI driver or the DB2 JDBC driver.

This information discusses the following topics:

- “Operating system or hardware maintenance”
- “Updating a DB2 CLI driver or a DB2 JDBC driver” on page 266
- “Rolling update of the DB2 CLI driver for Linux on z Systems and AIX” on page 266

Operating system or hardware maintenance

Any maintenance operation which requires a downtime of an individual SAP application server needs to be prepared by the SAP base administrator who should first move all SAP workload away from this server. This can be done, for example, by adjusting SAP logon group definitions or by switching the SAP operation mode via SAP transaction RZ03. As soon as there is no more SAP workload on the application server it can be stopped.

With System Automation for Multiplatforms:

Follow the procedure described in step **Stop application servers** of “Stopping an SAP system with System Automation for Multiplatforms and SA z/OS” on page 246.

When the maintenance operation is complete you can restart the SAP application server. To do this, follow the System Automation for Multiplatforms procedure as described in step **Starting SAP application servers** of topic “Stopping an SAP system with System Automation for Multiplatforms and SA z/OS” on page 246.

With SAP application servers managed remotely by SA z/OS:

To start or stop SAP application servers in SAP installations where the SAP application servers are controlled remotely by SA z/OS, refer to “Maintenance that requires a reboot of an SAP application server” on page 268.

Updating a DB2 CLI driver or a DB2 JDBC driver

Updating your system to a new DB2 Connect fix pack or to a new version of the DB2 Connect product includes:

- an update of the DB2 CLI driver for your SAP ABAP application server instances
- an update of the DB2 JDBC driver for your SAP Java application server instances.

It is not possible to update the DB2 JDBC driver without SAP Java application server downtime. Therefore, for Java instances, you must follow this standard procedure:

1. Rebind the DB2 packages for the new driver (if applicable).
2. Replace the DB2 JDBC driver files in the filesystem.
3. Restart the Java instances so that they pick up the new DB2 JDBC driver level. Follow the procedures in “Managing an SAP system” on page 243 for this restart. If you are running a dual-stack (ABAP and Java) SAP system then you need use the procedure to restart both the ABAP and the Java instance on each SAP application server.

To update the DB2 CLI driver you can use a *rolling update* procedure, which is applicable to Linux on z Systems and AIX application servers. This procedure allows you to update the driver without restarting the ABAP instances and without any service interruption to the end-user. For a description of this procedure, read “Rolling update of the DB2 CLI driver for Linux on z Systems and AIX.”

Rolling update of the DB2 CLI driver for Linux on z Systems and AIX

Complete the steps in the following procedure to update your DB2 CLI driver for the ABAP application servers instances with minimal impact on the operation of your SAP system. Binding the DB2 CLI driver to DB2 for z/OS is only necessary when a new major version of the DB2 CLI driver driver is installed. New fixpack levels, special builds, or minor version changes for the DB2 CLI driver do not require a re-bind.

Use **db2radm** to issue the re-bind in case it is required. The DBSL autobind feature that was recommended in earlier versions of this publication is no longer available with DB2 12.

Follow these steps to update the DB2 CLI driver:

1. **Obtain the new the DB2 CLI driver.** Check *SAP Note 1927404: DB2-z/OS: IBM Data Server Driver for CLI/ODBC/JDBC/SQLJ - Special Builds* for the DB2 CLI driver levels that are certified for SAP and obtain the new driver directly from the **SAP Software Download Center**.
2. **Update the DB2 CLI driver in the SAP global directory.**
 For AIX, replace the DB2 CLI driver in: /sapmnt/<sid>/global/db2/AIX_64.
 For Linux on z Systems, replace the DB2 CLI driver in: /sapmnt/<sid>/global/db2/LINUX_S390X_64.
3. **Install the DB2 Connect license.** Ensure that you have valid license installed for the DB2 CLI driver. For information see chapter *Obtaining Licenses for the CLI and JDBC Drivers* in the *Database Administration Guide: SAP on IBM DB2 for z/OS*. A new license is only required if there is a major version change in the DB2 CLI driver.
4. **Re-bind the DB2 CLI driver if there is a major version change.** Use **db2radm** to issue the re-bind in case it is required.
5. **Check if the DB2 CLI driver was successfully installed (and potentially re-bound) and that it can be accessed in the SAP global directory.** Execute the following command as <sid>adm user:

```
R3trans -d
```

The return code must be 0. The trans.log trace file shows the DB2 CLI driver level:

```
DB2 Connect driver version 11.01.0000
DB2 Connect driver build level is special_35734
```

6. **Activate the new DB2 CLI driver.** You usually do this by restarting one ABAP application server at a time. As a consequence, each application server has a short outage. During a (re)start of an application server instance, the sapcpe mechanism copies the new DB2 CLI driver from the global directory into the local instance directory (for details see the SAP online help and search for topic: *The sapcpe Program*). The restarted SAP work processes are then using the new level of the DB2 CLI driver.

 In order to avoid any disruption of service for your SAP system, you can use the new automated rolling kernel switch mechanism (for SAP Netweaver 7.4x and higher) to restart one ABAP application server instance at a time. (see "Updating the SAP kernel using the rolling kernel switch" on page 272).
7. **Check that the new version of the DB2 CLI driver is active for a re-started instance in the SAP developer trace files dev_w<n>.** Upon successful connect to the database these files show the CLI driver level in a message similar to this:

```
DB2 Connect driver version 11.01.0000
DB2 Connect driver build level is special_35734
```

Change management on SAP Central Services hosts

This topic describes several tasks that need to be done before you can safely apply service and maintenance to the z/OS operating system or to the hardware (LPAR) that runs your SAP central services. To keep your business operating continuously and secure, all SAP software should be stopped and/or moved away from the operating system and the LPAR that require service. This includes for example the SAP Central Services and the SAP control function for z/OS remotely managed SAP application servers.

At first, move the control function of an SAP application server managed remotely from SA z/OS.

The following command samples assume that you have implemented the *SAPSRV *add-on policy* for remotely managed SAP application servers. Logon to a NetView command session (SA z/OS command dialog) and list the remotely managed SAP AS application MOVE groups to get an overview. Issue **INGMOVE SAP<SID>RM***.

1. Find the MOVE group that is displayed with an observed status of AVAILABLE on the system that you would like to stop for maintenance. Move that group to a system from which you know it will stay up and available during the maintenance. A list of possible systems is displayed beneath the headline **System**.
2. Type a target system name into column **Move to**, then type m into the command column.
3. Press the Enter key and verify the selection. Then press PF10 to actually start the move.
4. Use PF9 to refresh the status until the move successfully completes. Now all resources used to control the remotely managed SAP AS should be off that system.
5. Verify the result with command **INGLIST SAP<SID>R***.

Now you can continue with other tasks for preparing the maintenance. After maintenance is complete, you can use the same procedure to move the control function back.

SAP tools and restarting SAP application server instances

SAP tools perform certain maintenance tasks on your SAP system which restart one or more SAP application server instances as part of their operation but at the same time leave the SAP central services and the DB2 database running.

An example of such a maintenance task is the operation of the SAP Software Update Manager (SUM).

If you use the following procedure, System Automation for Multiplatforms does not interfere with the restart operations that are performed by these SAP tools.

1. Start your SAP system with System Automation for Multiplatforms.
2. Set System Automation for Multiplatforms into manual mode with:
`samctrl -M T`

When System Automation for Multiplatforms is running in this mode, it monitors all resources in the cluster, but it does not take any actions to start or stop resources. You can use `lssamctrl` to check the actual mode.

3. Perform the maintenance activities.
4. After the maintenance is installed, set SAP into the automatic mode again:
`samctrl -M F`

System Automation for Multiplatforms recognizes whether the SAP tool did restart the SAP instance and if it did it displays the resource as ONLINE. If the SAP tool did not restart the SAP instance, System Automation for Multiplatforms tries to restart it according to its policy.

Maintenance that requires a reboot of an SAP application server

The topic discusses the maintenance of application servers that are controlled in different ways.

- SAP application servers that are controlled by SA MP
- SAP application servers that are controlled remotely by SA z/OS

SAP application servers that are controlled by SA MP

If you need to install other maintenance at a node while SAP is not running or you need to restart a node in your cluster, remove the node from the cluster by adding the node to the excluded node list:

```
samctrl -u a <nodename>
```

This command stops all fixed resources of the node and tries to move active floating resources to another node. Use the `lssamctrl` command to check whether a node is currently contained in the excluded node list of the cluster. If it is excluded and all resources are offline, you can apply your maintenance, restart, and so on. As long as you do not remove the node from the exclude list, or in other words join the node to the cluster again, System Automation for Multiplatforms does nothing to the node. Issue the following command to join the node to the cluster again:

```
samctrl -u d <nodename>
```

SAP application servers that are controlled remotely by SA z/OS

Management of SAP application servers (SAP AS) is slightly different when you use a configuration where the SAP application servers are remotely controlled by SA z/OS. The topic describes how to remotely stop and restart an SAP AS from SA z/OS. It is applicable for SAP application server maintenance or maintenance on the node, which itself hosts the SAP application server.

The following command samples assume that you implemented the **SAPSRV add-on policy* for remotely controlled SAP application server.

Log on to a NetView command session (SA z/OS command dialog) and list the remotely managed SAP AS resources to get an overview. Issue `INGLIST SAP<SID>R*`. Look for sysplex move groups `SAP<SID>RM*` and find the one that holds the SAP application server that you would like to stop for maintenance.

If you have changed or set up the remote application server policy to use the REXX script `SAPRASTP` in `SHUTNORM`, then a `STOP` vote on this move group with `TYPE=NORM` (default) is sufficient. `SAPRASTP` implicitly changes this to `TYPE=FORCE`.

Otherwise, set an explicit `STOP` vote on this move group with `TYPE=FORCE`. The `FORCE` actually stop the SAP application server. Without `SAPRASTP`, a stop with `TYPE=NORM` only stops the z/OS monitoring of the SAP application server. The SAP application server itself continues to run.

Wait until the sysplex move group reaches the observed status `SOFTDOWN`. Start to perform your maintenance tasks on the application server node. This might even include restarts of the SAP AS host system without interfering the status of the SAP AS.

As soon as your maintenance is completed, restart the SAP application server by canceling the `STOP` vote of the sysplex move group `SAP<SID>RM*`. Wait until the sysplex move group reaches the observed status `AVAILABLE`. The application server instance can now be used again for SAP business.

SAP Software Update Manager (SUM)

The SAP Software Update Manager (SUM) combines many SAP update and upgrade mechanisms that were implemented separately in previous SAP releases into one single tool. SUM can perform release upgrades, install enhancement packages, and install updates for single components of an SAP system.

SUM assumes and requires that your SAP central services are used SAPs sapstartsrv framework. Earlier versions of the SAP policy, for example the **SAP add-on policy* of SA z/OS 3.4 or 3.3, did not use sapstartsrv to start the central services. If you are still running such a policy, you should consider moving to the recommended **SAPSRV add-on policy*. This policy uses the sapstartsrv framework to start SAP central services. If you are not able to switch to the **SAPSRV add-on policy*, you might consider by using the approach that is described in **SAP Note 2380717: SUM in HA environments with *SAPSRV add-on policy**. Note that you cannot use the SAP HA Interface in this case and must switch off automation altogether during run time of SUM.

Before you start a SUM operation in an SAP high availability environment, you must carefully plan this change. The reason is that the SUM procedure involves certain steps (for example, stop and restart SAP instances). These SUM actions might interfere with the automation product that tries to keep the SAP components available always.

The simplistic approach to avoid any interference is to disable automation for all SAP components of the SAP system that SUM needs to stop or (re)start. The disadvantage of this approach is that for the complete elapsed time of the SUM operation you lose your capability to automatically recover from unplanned outages.

Therefore, the recommended way - which allows you to maintain the highest level of availability for your SAP components - is to use the SAP HA Interface as described in "SAP HA Interface for SA z/OS" on page 168. This interface enables seamless interaction between SAP's management actions (start or stop) and Tivoli System Automation for z/OS. The SUM processing has a downtime in the EXECUTION phase which can be reduced by selecting nZDM (near Zero Downtime Maintenance) or ZDO (Zero Downtime Option). Using the SAP HA Interface keeps your system highly available before and after this phase. Note that the SAP *shadow* and *upgrade* (ZDO) subsystems which SUM creates and uses internally, are not part of the automation policy. Therefore, SA z/OS does not start, stop, or monitor these subsystems.

Scenario 1: SAP central services that are automated via SA z/OS - no automation for SAP Application servers: If your SAP central services instances are automated via SA z/OS, then you only need to install and activate the SAP HA Interface for SA z/OS. During its operation, SUM performs a stop and a restart of the central services instance. The SAP HA Interface makes sure that SA z/OS votes are inserted to stop the instance and that the votes are removed afterward to restart the instance.

Scenario 2: - SAP central services and SAP application servers that are automated via SA z/OS: For the SAP central services, the same recommendation as in scenario 1 applies. For the SAP application servers, you have two options:

- Disable the automation for the SAP by setting the automation flag to NO for the SAP<SID>RMn_X resources. You lose the application server restart capability of

the SA z/OS policy during runtime of SUM, but this may be an acceptable risk. Make sure that you enable automation for the remote application server resource again after SUM has finished.

- Install the remote SAP HA Interface for SA z/OS. SA z/OS restarts application server instances if an unplanned outage is detected. When SUM restarts application server instances as part of its normal operation, then SA z/OS votes are inserted and removed again when restarting the instance. You must have installed and activated version 2.0 or higher of the remote SAP HA Interface for SA z/OS for all application servers.

Updating the SAP kernel

It is important for an SAP system that all application server instances use the same kernel level. For this reason SAP implemented a checking mechanism to ensure consistent kernels. In this section, this mechanism is described in detail to help you understand why kernel updates must 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 about the platform type (for example, Linux on z Systems, AIX, or Windows) 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.

An exception to this rule is the SAP rolling kernel switch mechanism (see “Updating the SAP kernel using the rolling kernel switch” on page 272), which tolerates different kernel levels on the application servers.

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.

If you are using the old central instance concept, this mechanism is very reasonable. The message server is started and stopped with the central instance.

The stored patch level is that of the central instance. However, with the new concept, the application server instances might connect in an arbitrary order. Furthermore, the instances are started and stopped independently of the message server. A new patch level for the application server instance (disp+work) usually does not require a new patch level of the message server or the enqueue server. Nevertheless it is recommended to keep the patch level of central services instance on a current level.

The SAP program `sapcpe` ensures automatic synchronization of centrally and locally installed executables. Starting an SAP instance, the program `sapstart` calls program `sapcpe`. During the system start, `sapcpe` copies the SAP kernel from the central exe directory (indicated by the profile parameter **DIR_CT_RUN**) to the instance-specific exe directories (indicated by the profile parameter **DIR_EXECUTABLE**). After installing a new SAP kernel in the central exe directory, you must restart all SAP instances that use this kernel.

For SAP systems where a downtime is acceptable, this can be done in a disruptive way, as described in “Updating the SAP kernel using a disruptive procedure.” For productive SAP systems or systems that require higher levels of availability, the preferred method, which avoids a complete SAP system downtime is described in “Updating the SAP kernel using the rolling kernel switch.”

The SAP rolling kernel switch procedure comes in two different flavors, depending on the SAP kernel level:

- With SAP 7.2x kernels, the rolling kernel switch needs some manual actions to be performed by the SAP administrator.
- With 7.4x kernels, the rolling kernel switch is available as automated procedure. The SAP administrator just starts the update procedure, which then updates the kernels of all SAP instances in the correct order.

Note: The rolling kernels switch procedures apply to ABAP-only instances only. For details on kernel compatibility considerations for Java application server instances see *SAP Note 953653: Rolling kernel switch*

Updating the SAP kernel using a disruptive procedure

If the SAP kernel (disp+work or one of its dynamically loaded modules) is to be updated, complete the steps that are documented in this topic.

The sequence is applicable for SAP application servers running on Linux or AIX:

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, see “Operating system or hardware maintenance” on page 265. Wait until all application servers are stopped.
3. Stop and restart the message server.

With System Automation for z/OS, this is accomplished by issuing a STOP vote for the SAP Central Services group with the RESTART option set to YES.

4. Start the application server instances again, see “Operating system or hardware maintenance” on page 265.

Note: On Windows, load modules cannot be replaced while they are in use. Therefore, first stop the application server instance before replacing the executable routines 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 SAP kernel using the rolling kernel switch

The SAP rolling kernel switch (RKS) is a method that allows you to upgrade your SAP kernel patch level while the SAP system stays available. Therefore, it is the preferred method for kernel updates in SAP systems that require the highest level of availability. A prerequisite for using a rolling kernel procedure is that the SAP central services run as separate instances. The older central instance concept is not supported.

A general overview of the procedure can be found in the SAP documentation Rolling Kernel Switch - Updating the SAP Kernel Without System Downtime. More details are contained in the central SAP note that describes prerequisites and the manual and automated variants of the rolling kernel switch: *SAP Note 953653: Rolling kernel switch*.

Manual rolling kernel switch with SAP kernels 7.2x

To avoid loss of enqueue locks during the RKS, this variant of the RKS requires that enqueue replication is active and that an SA z/OS policy for the SAP central services is active.

Enqueue replication must either be implemented running the central services with *EnqCF replication* (see Chapter 13, “Enqueue replication into a z Systems coupling facility,” on page 277) or with a separate enqueue replication server. Furthermore you need to ensure that the sapcpe mechanism is implemented as recommended by SAP.

Different SAP kernel levels can only be active in the same SAP system if they are compatible. SAP maintains the information which kernels are compatible in the (Statement of Compatibility) file StoC.xml, which is needed in the rolling kernel switch procedure (see *SAP Note 953653: Rolling kernel switch*).

If a kernel patch is compatible with its predecessor, the kernel patch can be installed and activated in a rolling fashion by first restarting the central services and then one application server instance after the other, rather than shutting and restarting the complete SAP system.

If SAP Central Services are managed by SA z/OS, then you should perform the RKS as follows:

1. Save the current kernel level(s).
2. Install the new kernel in the global SAP exe directory for the application servers and for the SAP Central Services.

For SA z/OS managed ASCS, then perform the following:

1. If you are running a separate ERS instance: Stop the ERS instance (enqueue replication server) in SA z/OS with the option RESTART = YES.
2. After successful completion, stop the ASCS instance (enqueue and message server) in SA z/OS with the option RESTART = YES.

Before restarting an SAP application server instance, you should first use SAP transactions to quiesce all SAP workload before restarting the instance. If an SAP application server instance is managed either via SA z/OS or via SA MP, use the automation software to stop and restart the instance

Note: If SAP instances have the SAP HA Interface activated (see “SAP HA Interface for SA z/OS” on page 168), the SAP administrator can use standard SAP mechanisms (SAP MC or the SAP command line tool sapcontrol) to stop and restart the instances.

Automated rolling kernel switch with SAP kernels 7.4x

With SAP kernels 7.4x, SAP has significantly enhanced the RKS procedure, which can now run in a fully automated fashion, requiring no manual input from the SAP administrator after it is started.

When the automated procedure stops and restarts all SAP instances in sequence, the central services instance is stopped and restarted first and afterward the application server instances are stopped and restarted one after another. For a flow diagram, see the SAP documentation that is referenced in *SAP Note 953653: Rolling kernel switch*.

With 7.4x, the compatibility level of different SAP kernel levels is no longer checked by using an external xml file, but is a property of the SAP kernel itself. The automated RKS procedure checks these levels as part of its operation. For prerequisites and procedures, refer to *SAP Note 953653: Rolling kernel switch* and the 7.4x SAP Help documentation that is referenced in this SAP note.

The automated RKS can only be started for an SAP system with a redundant setup: the system must have been installed with SAP Central Services and must be running with active replication. In addition, there must be at least two application server instances, which have all SAP services (spool, batch, update, dialog) to avoid a downtime while any of the instances is restarted.

Running the rolling kernel switch while automation is active

The rolling kernel switch (RKS) itself is capable of restarting SAP with a new kernel by smoothly restarting one SAP instance at a time without any automation solution being active. Therefore, one option is to switch off automation during RKS operations.

Nevertheless, you might want not to lose the failover capabilities that are offered by SA z/OS during RKS. The following paragraphs describe the conditions under which you can keep SA z/OS automation active while running RKS. See also *SAP Note 2131873: z/OS: Automated Rolling Kernel Switch in HA environment* for further prerequisites and restrictions.

RKS with automation for Central Services

If your SAP central services are automated via an SA z/OS policy, you can keep this policy active during the runtime of the RKS, if you provide for the following:

1. Use *EnqCF replication* with a matching SA z/OS policy (see “Option 2: Using EnqCF replication only” on page 280).
2. Use the HA interface (see “SAP HA Interface for SA z/OS” on page 168).

RKS restarts the SAP central services (including the SAP enqueue server) in-place. The SA z/OS **SAPSRV add-on policy* allows this restart in-place only if you use *EnqCF replication* and have no ERS instance configured.

Using the HA interface ensures that RKS commands to stop and start the Central Services are processed by SA z/OS.

If you use the traditional replication mechanism with a separate ERS instance, the SA z/OS policy does not allow a restart in-place. A restart of the ASCS instance takes place on the LPAR where ERS was running in order to avoid losing SAP enqueue locks. While this restart is active, RKS cannot run successfully. Therefore, you must disable automation before starting RKS. For this purpose, set the SA z/OS automation flag to N0 for the SAP<SID>ASCSX and SAP<SID>AER_X sysplex move groups. After RKS successfully finished, make sure that you reset the automation flags to YES again.

RKS with Automation for SAP application servers

If your SAP application servers are automated using SA z/OS (see “SAP application servers as proxy resources” on page 171) then you should take the following into account before running RKS.

SAP recommends that you should not have automation active for your SAP application server instances when running RKS (see *SAP Note 2077934: Rolling kernel switch in HA environments*).

One of the reasons for this recommendation is the fact that RKS uses specialized soft shutdown procedures for stopping the application server instances, instead of using the standard (hard) shutdown that the automation uses to stop them. So in this scenario you should disable automation for the SA z/OS proxy resources by setting the automation flag to NO for the sysplex move groups SAP<SID>RM0_X, SAP<SID>RM1_X, ..., and so on. After RKS successfully finished its operation and stopped and restarted the Application Server instances, you should reset the automation flag to YES. SA z/OS then recognizes that the application servers have been restarted already and the status of the SA resources are AVAILABLE.

If you have no special requirements for a soft shutdown, you can choose to keep automation for application server instances active during RKS. In this case, you should configure the HA interface for SAP application servers (see "SAP HA Interface for SA z/OS" on page 168).

If your SAP application servers are automated using SA MP (see Chapter 9, "Customizing System Automation for Multiplatforms," on page 185), you should set the automation to manual mode by using the `samctr1` command before starting RKS and reset it to automatic mode after RKS successfully finished.

Resolving PROBLEM / ZOMBIE states at restart of sapstartsrv

When RKS restarts SAP instances, then the new SAP kernel is copied from the global executable directory to the local instance executable directory. While the SAP instance is now started using the updated executables, the sapstartsrv process continues to run with the old level. In order to pick up the new code level, SAP has the following automatic restart mechanism that is built in to the sapstartsrv executable:

Whenever a running sapstartsrv detects that the file in the local instance executable directory from which it was started has changed, the running sapstartsrv process restarts itself after five minutes - using the new executable. This restart takes place in-place and is rather fast. Under certain timing conditions, SA z/OS 3.5 is not quick enough to detect this. As a consequence, the SAP<SID>A_SRV resource is first shown with an observed status of STOPPING and is later shown with a PROBLEM status (SA z/OS agent status is ZOMBIE). To resolve this, check that the z/OS UNIX sapstartsrv process is running. If so set the SA z/OS state of the resource to UP.

Chapter 13. Enqueue replication into a z Systems coupling facility

Enqueue replication is a high-availability concept that is built-in in SAP systems and that protects SAP enqueue locks in case of planned or unplanned outages of the SAP enqueue server or the host on which it is running. Besides the standard SAP enqueue replication, an additional replication mechanism is available for SAP on IBM z Systems. This topic describes the business continuity aspects of this additional replication mechanism, and how it can be integrated into an existing Business Continuity setup for SAP.

Starting with SAP 7.21 Kernel, the SAP enqueue server running on z/OS can utilize a new replication mechanism for its enqueue locks. For details on implementation, configuration, and requirements see *SAP Note 1753638: Enqueue Replication into System z Cross Coupling Facility* together with the attached PDF file.

The **standard SAP enqueue replication** operates in the following way:

- a replication server is started on a different host than the one where the enqueue server is running
- the enqueue server and the replication server communicate via TCPIP
- enqueue lock information is sent from the enqueue to the replication server
- if the enqueue server fails, it must be restarted on the host where the replication server was running
- the restarted enqueue server picks up the replication information and recovers the enqueue lock information

In contrast to the standard SAP enqueue replication, the new replication method that is available for SAP on z Systems does no longer require a replication server. Instead, the enqueue server stores its replication information directly into the cross-system coupling facility (XCF) storage which is accessible from all LPARs in a z/OS sysplex.

In this publication, this new replication mechanism is abbreviated as *EnqCF replication*, whereas the standard replication method is called *TCPIP-based replication*.

With *EnqCF replication*, the recovery scenario for an enqueue server failure changes as follows:

- Replication information is constantly and directly stored into the coupling facility (CF) by the SAP enqueue server.
- If the enqueue server fails, it can be restarted on any LPAR in the sysplex.
- The restarted enqueue server picks up the replication information from the CF and recovers the enqueue lock information.

Table 40. Comparison of **EnqCF replication** and **TCPIP-based replication**

Criteria	TCPIP-based replication	EnqCF replication
replication server	<ul style="list-style-type: none"> is an essential part of an high availability setup must be started on a host that is different from the one where the enqueue server is running 	<p>not needed:</p> <ul style="list-style-type: none"> no need for an automation policy which ensures the anti-collocation of enqueue and enqueue replication server
planned or unplanned failover of replication server	<ul style="list-style-type: none"> replication server must be restarted on a host that is different from the one where the enqueue server is running while the replication server is being restarted , a failure of the enqueue server cannot be recovered 	<p>not applicable:</p> <ul style="list-style-type: none"> no need to automate the replication server replication server no longer has any impact on the availability characteristics of the enqueue server
planned restart of enqueue server	<ul style="list-style-type: none"> enqueue server must be restarted on LPAR where replication server was running to avoid loss of enqueue locks replication server must be moved to a different LPAR 	<p>more flexible:</p> <ul style="list-style-type: none"> enqueue server can be restarted on any LPAR in the sysplex, even on the same LPAR SAP Central Services restart is possible without loss of enqueue information even with only one LPAR being active
unplanned failover of enqueue server	<ul style="list-style-type: none"> enqueue server must be restarted on LPAR where replication server was running to avoid loss of enqueue locks replication server must be moved to a different LPAR (if available) 	<p>more flexible:</p> <ul style="list-style-type: none"> enqueue server can be restarted on any LPAR in the sysplex no sophisticated automation policy required which ensures that enqueue server is started on the correct LPAR

High availability considerations for EnqCF replication

The **SAPSRV add-on policy* included with SA z/OS contains resources and relationships that model the SAP enqueue and SAP enqueue replication server on z/OS, based on standard *TCPIP-based replication*. When creating a system automation policy for an SAP system that is running with the **SAPSRV add-on policy*, you may want to adapt this policy. The required adaption depends on how you plan to operate your SAP central services on z/OS. Two main options for adapting the policy are discussed in this topic.

The two main options for adapting the **SAPSRV add-on policy* for *EnqCF replication* are:

- Option 1:** maintain the ability to switch back and forth between *TCPIP-based replication* and *EnqCF replication*
- Option 2:** permanently and exclusively switch to the new *EnqCF replication* mechanism

The first option might be suitable for:

- the implementation phase where you introduce the new replication mechanism into existing SAP production systems
- SAP test systems.

The second option might be more suitable for newly installed SAP production systems that require the highest level of availability and which will permanently use the new replication mechanism.

Topics “Option 1: Maintaining switchover capability between TCPIP-based and EnqCF replication” and “Option 2: Using EnqCF replication only” on page 280 describe the changes that are required in the System Automation policy for these options. It is assumed that you are running with an SA z/OS policy that is based on the **SAPSRV add-on policy*.

Note: Make sure that you do not switch on *System Managed Duplexing* for the XCF structures that are needed for *EnqCF replication*. Duplexing is not necessary and in addition has a negative impact on the performance of the replication.

Option 1: Maintaining switchover capability between TCPIP-based and EnqCF replication

The SA z/OS policy must have been generated - preferably using the SAP HA wizard - with full support for SAP enqueue replication server resources. These resources are then modelled in the following SA groups in your policy (<SID> is the placeholder for your 3-letter SAP system ID).

For SAP ABAP central services the SA groups are:

- SAP<SID>AER_X
- SAP<SID>ARSRVX [automation name S<SID>ARSRV_X]

For SAP Java central services the SA groups are:

- SAP<SID>JER_X
- SAP<SID>JRSRVX [automation name S<SID>JRSRV_X]

When running your SAP central services with *TCPIP-based replication*, you need to make sure that these groups and their resources are being started by the active policy to ensure that enqueue replication is possible.

As soon as you switch your SAP system to *EnqCF replication*, then these groups and the contained resources are no longer required, since *EnqCF replication* does not need a replication server. See *SAP Note 1753638: Enqueue Replication into System z® Cross Coupling Facility* and the attached PDF file for details on how to implement the switch.

Switching the replication mechanism to *EnqCF replication* involves a downtime of your central services instance(s) and should preferably be implemented during a planned downtime of your SAP system.

If you restart the SAP central services after switching to the *EnqCF replication* mechanism without making any changes to your existing System Automation policy, the SAP enqueue replication server is started by the policy although it is no longer needed. The replication server attempts to connect to the enqueue server, but the connection attempts will never be successful, because an enqueue server, that is set up for *EnqCF replication*, does not accept any (TCPIP) connection requests from the replication server.

The log file of the enqueue replication server *dev_enqsrv* shows the following connection errors for every failed connection attempt. As a default, the replication server makes a connection attempt every 20 seconds:

```

[Thr 21140C00:00000000] Fri May 24 08:08:05 2013
[Thr 21140C00:00000000] RepServer: main: Reset the transaction stamp to 0 0 0
[Thr 21140C00:00000000] ***LOG Q0I=> NiPConnect2: 10.101.5.194:6500: connect (1128:
EDC8128I Connection refused.) [./nixxi.cpp 3286]
[Thr 21140C00:00000000] *** ERROR => NiPConnect2: SiPeekPendConn failed for hd1 7/sock 7
(SI_ECONN_REFUSE/1128; I4; ST; 10.101.5.194:6500) [nixxi.cpp 3286]
[Thr 21140C00:00000000] *** ERROR => EncNiConnect: unable to connect (NIECONN_REFUSED)
[encomi.c 447]
[Thr 21140C00:00000000] *** ERROR => RepServer: main: connect failed with return code -7
[enrepserv.cp 750]

```

Although the replication server is running in this connection retry loop, System Automation nevertheless shows the replication server resources with status AVAILABLE. Furthermore, the enqueue server resource still follows the enqueue replication server resource. This means, that whenever a planned or unplanned failover is triggered, the enqueue server is restarted on that LPAR where the replication server was previously running, although this is technically no longer necessary. With *EnqCF replication*, the enqueue server can be started on any available LPAR in the z/OS sysplex. Furthermore, the consequence behavior described in “Implementation of Dependency 3” on page 163 still applies.

If you want to avoid these limitations together with the error messages in the replication server log file, you should stop the replication server groups, for example, by defining **SA z/OS service periods** or issuing STOP votes for the System Automation groups.

Make sure you remove the STOP votes (or modify the service periods) before switching back to *TCPIP-based replication*.

Option 2: Using EnqCF replication only

When creating a new System Automation policy using the SAP HA wizard, you can remove the System Automation resource groups that contain the replication server. During its interactive operation, the SAP HA wizard asks the user if the replication server resources should be removed:

```

The SAP ABAP enqueue replication server resources can be removed,
if SAP local replication into the coupling facility is used.
Should the following Enqueue Replication Server resources
be deleted from the generated policy (y/n)?

```

If you answer Y to this question, then the resources and groups are removed from the resulting policy. This SA z/OS policy can be used with *EnqCF replication*. It no longer contains any replication server resources and the enqueue server can be started or restarted anywhere in your z/OS sysplex.

Note: The generated policy does no longer work for *TCPIP-based replication*.

As an additional change, you might consider changing the critical threshold for the enqueue server resource SAP<SID>AEN (or SAP<SID>JEN). This threshold is set to 1 in the **SAPSRV add-on policy*. A critical threshold of 1 is absolutely required for *TCPIP-based replication* such that SA z/OS never tries to restart the enqueue server in place. Instead, the enqueue server is restarted on a different LPAR (than the one where the replication server was running), whenever the enqueue server terminates with an error.

For *EnqCF replication*, this behavior is no longer required. If you want to allow a restart in place, then you need to make the following changes in the enqueue server resources:

1. Increase the CRITICAL THRESHOLD value, for example, to 2.
2. Change the RESTART OPTION of the SAP<SID>AEN (or SAP<SID>JEN) resources from NEVER to ALWAYS.
3. In the SAP profile for the SAP central services instance, change the start mechanism of the enqueue server from starting to restarting, that is, change the SAP profile entry from :

```
Start_Program_01 = local $(_EN) pf=$(_PF)
```

to

```
Restart_Program_01 = local $(_EN) pf=$(_PF)
```

Tool support for EnqCF replication

For the *EnqCF replication* solution, SAP ships the `cleanrepstz0SCf` tool together with the SAP central services executables for z/OS UNIX. With the help of various scenarios, this topic discusses how this tool can support you to clear the contents of the replication store.

The purpose of the `cleanrepstz0SCf` tool is similar to the SAP provided `cleanipc` command. `cleanipc` allows the SAP administrator to remove any shared memory that is associated with a specific SAP system and a specific SAP instance. For the SAP central services, the `cleanipc` command deletes any shared memory that was used by the replication server to hold the replication table. This tool should be used with care and only in special circumstances, because deleting the shared memory prevents a newly started enqueue server from reading it and finding any replication records.

`cleanrepstz0SCf` has the same effect: it clears the contents of the replication store by effectively deleting the XCF note pad that holds the replication information of a specific SAP central services instance. Like `cleanipc`, `cleanrepstz0SCf` must be run as user `<sid>adm`.

Sample scenario – freeing up space in CF

An SAP system uses the coupling facility for replication, but at a later point in time, this SAP system is stopped and no longer used. The XCF note pad structure for this SAP system still exists and has a certain memory footprint inside your CF structure. You can use the `cleanrepstz0SCf` command in this case to remove the XCF note pad structure from the CF, thereby freeing up memory in the CF.

Sample scenario – upgrading your SAP system

This scenario assumes that you have upgraded your SAP system to a higher release level or you have exchanged your SAP kernel level with an incompatible new version. Before you start up the new SAP release for the first time, you should use the `cleanrepstz0SCf` command to make sure that any old replication information is deleted, and can no longer be picked up by the restarted enqueue server that runs on a higher SAP release or a higher (incompatible) kernel level.

Note: As an alternative to `cleanrepstz0SCf`, the z/OS operator can use the z/OS XCF utility `IXCDELNP` to delete an XCF note pad. When using this utility, the z/OS operator should be aware of the SAP naming convention for XCF note pads to avoid deleting XCF note pads which contain valid replication information. SAP's naming conventions for XCF note pads are described in the *SAP Enqueue Replication into System z Coupling Facility - Installation and Setup* document that is

attached to *SAP Note 1753638*. For a detailed description of the IXCDELNP utility, see the IBM documentation *MVS Setting Up a Sysplex*.

Troubleshooting for replication problems

With *EnqCF replication* enabled, problems can occur during the startup of the enqueue server, or the server can encounter replication problems at run time. This topic discusses how to handle such problems.

In addition to troubleshooting guidance for replication problems, this publication also provides information on failover and recovery if you use *EnqCF replication* in "Failover and recovery of SAP Central Services using EnqCF replication" on page 73.

Problems during enqueue server startup

An enqueue server that is started with *EnqCF replication* might encounter problems when accessing the z/OS coupling facility. The reaction to such errors depends on the underlying scenario. This topic discusses the following situations:

- **Scenario 1:** enqueue server cannot access the CF
- **Scenario 2:** enqueue server can access the CF, but during startup does not find any replication information in the CF. Creation of the new XCF note pad fails.
- **Scenario 3:** enqueue server can access the CF, and there is valid replication information in the CF. However, creation of the new XCF note pad fails.

Scenario 1 – no access to CF:

The enqueue server terminates with an error message and System Automation tries to restart the enqueue server on a different LPAR.

Problem resolution: If the underlying problem cannot be resolved fast enough, and you need your SAP system up and running, then a short-term solution might be to change the SAP profile such that replication is disabled. But this solution has the obvious drawback that any failures of the enqueue server might lead to a loss of SAP enqueue locks. Therefore, implement a permanent solution to replace this one-time solution.

Scenario 2 – no replication information available:

This scenario might occur in the following situation:

- The enqueue server is started with *EnqCF replication* enabled.
- It **does not** find any replication information in the CF. This is for example the case when:
 - *EnqCF replication* is turned on for the first time.
 - The CF structures were deleted before starting the enqueue server (by using the `cleanrepstz0SCf` tool or by entering original z/OS XCF commands).

If the enqueue server now encounters an error while creating the XCF note pad that is required for replication:

- An error message is written into the SAP developer trace file `dev_enqrepl`.
- The SAP enqueue server stops.
- System Automation detects this termination and tries to start the enqueue server on a different LPAR (or attempts a restart in place).

If this is a general problem (for example, with the CFRM policy, or an authorization problem), then all enqueue server resources go into a HARDDOWN status.

Problem resolution: The problem may be caused by wrong structure definitions in the CFRM policy which defines the CF structures and their sizes. For definition and sizing of CF structures, see the PDF file that is attached to *SAP Note 1753638*. If the underlying problem (for example, wrong CFRM policy) cannot be resolved fast enough, then a short-term solution might be to change the SAP profile such that replication is temporarily disabled.

Scenario 3 – with replication information available:

This scenario may occur in the following situation:

- The enqueue server is started with *EnqCF replication* enabled.
- It **does** find valid replication information in the CF, but it is not able to create the new XCF note pad, for example because of:
 - changes in enqueue table size which requires a larger CF structure than defined in the current CFRM policy
 - CF structures being deleted or redefined.

The most likely reason for these errors are SAP profile parameter changes. If you, for example, increase the SAP enqueue table size parameter *enque/table_size* in such a way that the CF structures are no longer large enough to hold the replication information for the new enqueue table size, then the (re)start of your enqueue server with *EnqCF replication* enabled will fail. The enqueue server does not terminate in this case, so that the enqueue locks, encountered in the old CF structure at start up, are not lost. Instead, the enqueue server continues, with replication being temporarily disabled. Messages indicating this situation are written into the SAP developer trace file *dev_enqrepl*. The enqueue server tries to re-establish replication into CF at intervals defined via the SAP profile variable *enque/enrep/stop_timeout_s*.

The default for this parameter is 300 seconds. You may want to set this timeout to a lower value to allow for a quicker restart of the replication mechanism after the cause for the CF problem has been resolved.

Problem resolution: The problem should be solved by defining a CF structure large enough such that the replication into CF is enabled again.

Replication problems at runtime

Like in traditional *TCPIP-based replication*, any disruption of the replication is not visible in System Automation unless you are using the monitor mechanism described in “Monitoring the health state of SAP enqueue replication” on page 166. The enqueue server resources continues to be shown with status AVAILABLE. Messages indicating the failing replication are written to the SAP developer trace file *dev_enqrepl*. Like *TCPIP-based replication*, *EnqCF replication* tries to re-establish the replication into the coupling facility at regular intervals. The interval length is specified by the SAP profile variable *enque/enrep/stop_timeout_s*.

You should consider setting the value for *enque/enrep/stop_timeout_s* lower than the default of 300 seconds. This enables a quicker restart of the replication mechanism after the cause for the CF problem has been resolved.

With TCPIP-based replication, the total number of retries is limited through the SAP profile variable: *enque/enrep/stop_retries*. When this limit is reached, the enqueue server stops the attempts to re-establish replication until the replication server is restarted.

With *EnqCF replication*, the value of the *stop_retries* profile variable specifies the number of attempts to reuse the old CF note pad structure. When this number of retries is exhausted, the enqueue server tries to create a new CF note pad structure. The recommendation is to leave this variable at its default value of 1 to allow for fast recovery.

Chapter 14. Reference of the z/OS high availability scripts

This information unit describes all scripts mentioned in this current edition of this publication. For each script, its purpose, the invocation syntax, and an explanation of the parameters is provided. Also, you can read how to obtain the scripts.

The *automation scripts* are needed for the operation of the **SAPSRV add-on policy* of SA z/OS. They are contained in the `ING_sap.tar` file that is shipped with the SA z/OS product in the z/OS UNIX directory `/usr/lpp/ing/SAP` (see “Automation scripts”).

The *sample scripts* which are mentioned in this publication (see “Sample scripts” on page 290) are provided on an as-is basis in file `zSAP_BusinessContinuitynn.zip`. You can download this file from the **Files** section of developerWorks: SAP on IBM z Systems Community:

`zSAP_BusinessContinuitynn.zip`

Automation scripts

The automation scripts are provided within the `ING_sap.tar` file as part of the SA z/OS product.

Important: The **SAPSRV add-on policy* expects the z/OS UNIX scripts to be located in the home directory of the `<sid>adm` user.

The information contained in this topic provides the syntax description and the explanation of the parameters for each available script. When called without any parameters, all scripts print out usage and version information. The SAP policy that is generated by the SAP HA wizard from the **SAPSRV add-on policy* calls the scripts with parameters that are adapted to a specific SAP environment. When you use the scripts for a manually created SA z/OS policy for SAP, you need to adapt the parameters according to your SAP system.

Required SSH setup

The scripts `start_as` and `stop_as` use SSH to access remote application servers on Linux, AIX, or Windows platforms from z/OS UNIX System Services. In order to avoid interactive password authentication, you must generate a public or private key pair with `ssh-keygen` on z/OS UNIX and transfer the public key to the `authorized_keys` file on the remote application server platform. This key exchange must be done for every z/OS UNIX host where the `start_as` or `stop_as` scripts may run. This exchange must also be carried out for every remote SAP application server that is involved.

Make sure that the remote (virtual) host name is contained in the `~/.ssh/known_hosts` file of the `<sid>adm` user in z/OS UNIX. This can be done by manually executing one sample `ls` command via SSH, checking the fingerprint of the remote host and answering the interactive question with `yes` as follows:

```
boecoh1> ssh ihlscoh4v ls
The authenticity of host 'ihlscoh4v (10.101.4.217)' can't be established.
ECDSA key MD5 fingerprint is bb:4e:1d:e1:63:75:73:c1:04:17:30:d1:9f:e3:33:c3.
Are you sure you want to continue connecting (yes/no)? yes
FOTS2274 Warning:
Permanently added 'ihlscoh4v,10.101.4.217' (ECDSA) to the list of known hosts.
```

```

bin
public_html
...
...

```

Required changes to protected web methods of sapstartsrv

The scripts use sapcontrol functions like GetVersionInfo, GetProcessList, and J2EEGetProcessList, for example to monitor the status of a remote SAP application server. Prior to SAP NetWeaver kernel 7.38, these functions are not protected and can be called without user authentication. The default set of unprotected functions has been changed with NetWeaver kernel 7.38 and later kernels. In order to use the function J2EEGetProcessList with later kernels without authorization, you need to apply one of the following settings in the profile of each Java application server:

- service/protectedwebmethods=SDEFAULT -J2EEGetProcessList -GetVersionInfo
- service/protectedwebmethods=DEFAULT

The following scripts are currently provided within the ING_sap.tar file as part of the SA z/OS product:

Table 41. Scripts for the ***SAPSRV add-on policy** of SA z/OS

Filename	Description
Scripts available for explicit invocation with syntax documented at the end of this table	
start_as	Sample shell script used to start a remote application server instance (ABAP only, dual-stack, and Java only).
stop_as	Sample shell script used to stop a remote application server instance (ABAP only, dual-stack, and Java only).
check_as	Sample script used to start the SAP monitor for remote application servers (ABAP only, dual-stack, and Java only).
start_cs	Shell script used to start the: <ul style="list-style-type: none"> • ABAP SAP Central Services (ASCS) instance • enqueue replication server instance belonging to the ASCS • Java SAP Central Services (SCS) instance • enqueue replication server instance belonging to the Java SCS • SAP Web Dispatcher instance • SAP Solution Manager Diagnostics Agent (SMDA)
start_sapsrv	Shell script used to start the SAP Instance Service sapstartsrv for ASCS, SCS, ERS, WD, and SMDA instances.
checkwd	Sample script used by the *SAP policy in SA z/OS to check the health of the SAP Web Dispatcher
Scripts internally used by other scripts or by the *SAPSRV add-on policy	
samsapctrl_asping	Sample shell script to monitor the status of application server (ABAP only, dual-stack, and Java only). Used internally by check_as.
sapolicy.sap.map_USS	The samsapctrl_asping script needs this message file. It must be installed in the same directory as the samsapctrl_asping script, namely in the home directory of the <sid>adm user.
sap_xplatform	Cross platform library of functions, internally used by start_cs, start_sapsrv, and samsapctrl_asping scripts.
SAPWDMTR	Sample REXX program to check, if the health check z/OS UNIX process wdi spmon for a specific Web Dispatcher instance is running. Used internally in the monitor resource for the SAP Web Dispatcher in the *SAPSRV add-on policy .

Table 41. Scripts for the ***SAPSRV add-on policy** of SA z/OS (continued)

Filename	Description
SAPRASTP	Sample REXX program to eliminate for an SA z/OS operator the usage of STOP FORCE mode when stopping a remote application server. Used internally in the monitor resource for the SAP Web Dispatcher in the *SAPSRV add-on policy .

start_as

This script is used to start a remote application server instance.

It takes the host name, the instance directory, the instance type of the application server, a job name, a number of maximum retries, and optionally the remote execution type, as parameters:

```
start_as <hostname> <InstDir> <InstType> <jobname> <maxretries> [<via>]
```

<hostname>

Name of the host where the SAP application server runs. In installations with virtual host names the parameter **<Hostname>** refers to the virtual host name of the application server system.

<InstDir>

Instance directory of the remote application server. Together with the host name, it identifies the instance. For example, if the application server is a secondary application server, and uses instance number 66, then the instance directory is named D66.

<InstType>

Specifies, which application server instance type is started. A value of 0 starts an ABAP only application server(AS), a value of 1 starts a dual-stack AS, and a value of 2 starts a Java only application server.

<jobname>

Unique jobname for the start_as shell script. You must define a jobname which is different from the jobname listed under Application Information of the SA z/OS proxy resource. This is required to avoid a false SA z/OS reaction on the Pre-process termination exit when the script ends.

<maxretries>

Parameter to configure how often the script checks if a starting application server comes up or not. Time between retries is 10 sec. If the first check is unsuccessful, the script retries for 10 * <maxretries> seconds.

<via> Optional parameter to identify the remote execution type (SSH) used to send commands to remote application servers (running under AIX, Linux on z Systems, Linux on System x, or Windows). If this parameter is not set, it defaults to SSH.

stop_as

This script is used to stop a remote application server instance, an ABAP only, a dual-stack, or Java only instance.

It takes the host name, the instance number, the instance directory of the application server, and optionally the remote execution type, as parameters:

```
stop_as <hostname> <InstDir> [<via>]
```

<hostname>

Name of the host where the SAP application server runs. In installations

with virtual host names, the parameter **<Hostname>** refers to the virtual host name of the application server system.

<InstDir>

Instance directory of the remote application server.

<via> Optional parameter to identify the remote execution type (SSH) used to send commands to remote application servers (running under AIX, Linux on z Systems, Linux on System x, or Windows). If this parameter is not set, it defaults to SSH.

check_as

This script is used to start the monitor for a remote application server instance.

It takes the host name and the instance number of the application server as parameters:

```
check_as <hostname> <InstDir> <InstType>
```

<hostname>

Name of the host where the SAP application server runs. In installations with virtual host names the parameter **<Hostname>** refers to the virtual host name of the application server system.

<InstDir>

Instance directory of the remote application server. Together with the host name it identifies the instance. For example, if the application server is a secondary application server and uses instance number 66, then the instance directory is named D66.

<InstType>

Specifies, which application server monitor is started. A value of 0 starts an ABAP only application server (AS) monitor, a value of 1 starts a dual-stack AS monitor, and a value of 2 starts a Java only AS monitor.

start_cs

The version 3.1 of this script is used to start an SAP Central Services instance, a replication server instance, a Web Dispatcher instance, or an SAP Solution Manager Diagnostics Agent instance via `sapcontrol` under z/OS UNIX, if the corresponding `sapstartsrv` service is running on the system. This means, it can be used to start an ASCS, SCS, ERS, WD, or SMDA instance. It uses the *sap_xplatform* library.

It is shipped with System Automation for z/OS in file `ING_sap.tar` in `/usr/lpp/ing/SAP`. The script contained in the tar file is required by and matches the **SAPSRV add-on policy* of SA z/OS.

The syntax for invoking `start_cs` is:

```
./start_cs <sid> <SCS/ERS/WD/SMDA_instance> <hostname> <jobname> <maxretries>  
<sid> SAP system identifier to which the SCS/ERS/WD/SMDA instance belongs.
```

<SCS/ERS/WD/SMDA_instance>

Instance name of component. This is the instance followed by the instance number, for example, ASCS20 for the ABAP central services instance with instance number 20.

<hostname>

Virtual host name of the SCS/ERS/WD/SMDA instance. The host name is used to identify the appropriate SAP profile for the instance.

<jobname>

Jobname, which is defined in the SA resource definition for the `sapstart`

process of the instance. The `start_cs` script runs under a different jobname and it uses this parameter to issue to the console the status of the real SA resource via BPXF024I message, like for example, BPXF024I (HA2ADM) SHA2AST ACTIVE.

<maxretries>

Parameter to configure how often the script checks if a starting SCS/ERS/WD/SMDA instance comes up or not. Time between retries is 10 sec. If the first check is unsuccessful, the script retries for 10 * <maxretries> seconds.

Note: If the start of a specific SCS/ERS/WD/SMDA instance takes more than the default of 90 seconds, then you must increase this value. At the same time you must adapt the *Start Delay* policy of the System Automation APL. For every additional retry, add 10 seconds to the two-minute default of the *Start Delay* policy.

Example:

```
start_cs HA2 ASCS20 ha2ascsv SHA2AST 9
```

start_sapsrv

The version 3.1 of this script is used to start an SAP instance service `sapstartsrv` via `sapcontrol` under z/OS UNIX. It uses the `sap_xplatform` library.

This means it can be used to start a `sapstartsrv` service process for an ASCS, SCS, ERS, WD, or SMDA instance.

The syntax for invoking the `start_sapsrv` script is:

```
./start_sapsrv <sid> <SCS/ERS/WD/SMDA_instance> <hostname> <jobname> <maxretries>
```

<sid> SAP system identifier to which the SCS/ERS/WD/SMDA instance belongs.

<SCS/ERS/WD/SMDA_instance>

Instance name of component. This is the instance followed by the instance number, for example, ASCS20 for the ABAP central services instance with instance number 20.

<hostname>

Virtual host name of the SCS/ERS/WD/SMDA instance. The host name is used to identify the appropriate SAP profile for the instance.

<jobname>

Job name, which is defined in the SA resource definition for the `sapstartsrv` process of the instance. The `start_sapsrv` script runs under a different job name and it uses this parameter to issue to the console the status of the real SA resource via BPXF024I message, like for example, BPXF024I (HA2ADM) SHA2ASR ACTIVE.

<maxretries>

Parameter to configure how often the script checks if a starting SAP instance service comes up or not. Time between retries is 10 sec. If the first check is unsuccessful, the script retries for 10 * <maxretries> seconds.

Example:

```
start_sapsrv HA2 ASCS20 ha2ascsv SHA2ASR 9
```

checkwd

This script checks if the SAP Web Dispatcher is running.

It uses the SAP `wdispmon` executable to continuously check that the SAP Web Dispatcher functions are available and returns with an error indication if it is no longer the case. The call syntax for this script is:

```
checkwd <LPAR_Name> <WD_SID> <WD_InstanceName> <VIPA_Name>
```

<LPAR_Name>

Name of the LPAR where the Web Dispatcher is running.

<WD_SID>

SAP system identifier with which the SAP Web Dispatcher was installed.

<WD_InstanceName>

Instance name of the Web Dispatcher installation. It is assumed that there is one Web Dispatcher instance, which can run on every LPARs in the SYSPLEX.

<VIPA_Name>

External host name under which the SAP Web Dispatcher VIPA is reachable.

Sample scripts

In order to use the *sample scripts*, you need to extract them from the `zSAP_BusinessContinuitynn.zip` file and transfer them to the operating system platform where you want to use them.

This *zip file* contains:

- **Readme:**

- **readme_v32**

- a `readme` file for the current version of `zSAP_BusinessContinuity32.zip`

- **z/OS UNIX utilities:**

- **GetFileSystemOwnership**

- Sample shell script to move file system ownership to a specified LPAR. Not needed, if running under z/OS V1R13 and sysplex-aware file systems are used.

- **INGNFSGS**

- Sample started task INGNFSGS to call `GetFileSystemOwnership` from an SA z/OS resource

- **sanchkv1.txt**

- Sample REXX program to check for and clear Move Group EXCLUDEs or AVOIDs

- **db2_sazoscmd.rex**

- REXX script that enables the SAP host agent on z/OS with its subcomponent `sapdbctrl` to interact with SAP databases (DB2 subsystems on z/OS) that are automated via a SA z/OS policy. For details see *SAP Note 1887279: DB2-z/OS: sapdbctrl: Prerequisites and Configuration*.

- **db2_sazosmap.rex**

- REXX script that is called by `db2_sazoscmd.rex` and implements the mapping between DB2 subsystem IDs (SSIDs) and System Automation for z/OS SYSPLEX MOVE group names.

- **Linux utilities:**

testNFS

Linux sample shell script used to wait up to 3 minutes (default) for ospf/zebra to consolidate the routing table

beforeafs.service

Linux SLES 12 sample service to test z/OS NFS server availability before starting the autofs service

- **AIX utilities:**

rc.local

AIX sample shell script used to wait up to three minutes (default) for gated to consolidate the routing table.

Chapter 15. Sample network setup and miscellaneous migration considerations

This topic contains miscellaneous information in two subtopics. The first subtopic outlines a highly available network that was part of a test implementation of a business continuity solution for SAP on DB2. The second subtopic presents migration hints and tips on NFSv4 migration and SAP System Automation policy migration, and also presents helpful hints and tips on Linux for SLES 11 and SLES 12.

This topic presents helpful information on a sample network configuration and discusses migration hints and tips in the following subtopics:

- “Network setup”
- “Migration considerations” on page 302

Network setup

Figure 58 shows the test setup:

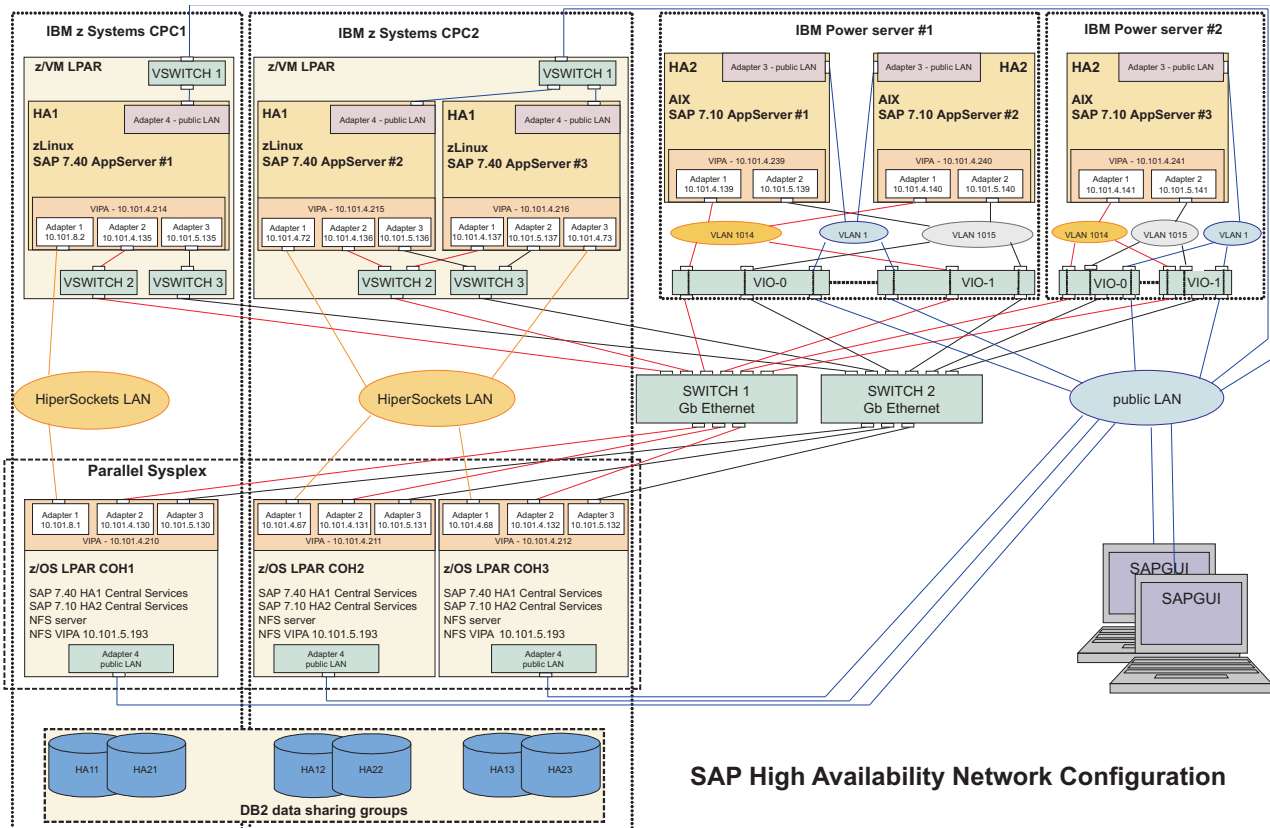


Figure 58. Networking configuration for the high availability solution for SAP

Network hardware components for the test setup

This information lists the network hardware used in the test setup to run the verification of the network availability.

The following hardware is used:

- Gigabit Ethernet adapters which are shared between LPARs
- HiperSockets are used as the preferred network connection for network traffic between systems that reside on the same hardware. The HiperSockets provide a third private network and a fast path between the HA1 Linux on System SAP application servers and the database servers.
- Two network switches for separation and true redundancy of the two private networks on Gigabit Ethernet.

Network software components for the test setup

This topic describes the following components and settings:

- “z/OS network settings”
- “Linux on z Systems network settings” on page 298
- “AIX OSPF definitions for the gated daemon” on page 299
- “Static VIPA definitions required for SUSE” on page 300

z/OS network settings

This information discusses various aspects of the required network setup of the z/OS network.

z/OS VIPAs

Dynamic VIPAs are defined as MOVEABLE DISRUPTIVE VIPAs for:

- SAP central services
- NFS server
- SAP network interface router (SAProuter)

z/OS UNIX System Services setup - BPXPRMxx

The following is an extract of the BPXPRMxx parmlib member used on all LPARs. It shows network definitions for the TCP/IP stacks and NFS definitions. For further details refer to *Planning Guide for SAP on IBM DB2 for z/OS*.

```
...
...
USERIDALIASTABLE('/etc/ualiastable')
...
...
FILESYSTYPE TYPE(NFS) /* NFS */
                ENTRYPOINT(GFSCINIT)
                PARM('DISABLELLA(y)')
                ASNAME(MVSNFSCS,'SUB=MSTR')
FILESYSTYPE TYPE(ZFS) /* ZFS */
                ENTRYPOINT(IOEFSCM)
                ASNAME(DFSZFS,'SUB=MSTR')
NETWORK        DOMAINNAME(AF_UNIX)
                DOMAINNUMBER(1)
                MAXSOCKETS(10000)
                TYPE(IBMUDS)
NETWORK        DOMAINNAME(AF_INET)
                DOMAINNUMBER(2)
                MAXSOCKETS(64000)
                TYPE(INET)
                RESOLVER_PROC(RESOLVER)
NETWORK        DOMAINNAME(AF_INET6)
                DOMAINNUMBER(19)
                MAXSOCKETS(32000)
                TYPE(INET)
```



```

MOUNT FILESYSTEM('OMVS.ZFS.COHPLEX.SAPMNT')
    MOUNTPOINT('/sapmnt')
    TYPE(ZFS)
    MODE(RDWR)
    AUTOMOVE
MOUNT FILESYSTEM('OMVS.ZFS.COHPLEX.USRSAP')
    MOUNTPOINT('/usr/sap')
    TYPE(ZFS)
    MODE(RDWR)
    AUTOMOVE
MOUNT FILESYSTEM('OMVS.ZFS.COHPLEX.TRANS')
    MOUNTPOINT('/usr/sap/transA11')
    TYPE(ZFS)
    MODE(RDWR)
    AUTOMOVE
MOUNT FILESYSTEM('OMVS.ZFS.COHPLEX.HA2.USRSAP')
    MOUNTPOINT('/usr/sap/HA2')
    TYPE(ZFS)
    MODE(RDWR)
    AUTOMOVE
...
...

```

z/OS COH1 – Unix System Services network settings

This section describes the network settings for LPAR COH1. The settings are also used for LPAR COH2 and COH3 with LPAR specific settings changed accordingly.

File /etc/resolv.conf

```

DATASETPREFIX SYS1.TCP
TCPIPJOBNAME TCPIP
BOECO1: HOSTNAME coh1vipa
BOECO2: HOSTNAME coh2vipa
BOECO3: HOSTNAME coh3vipa
DOMAINORIGIN boeblingen.de.ibm.com
LOADDBCSTABLES SJISKANJI
NSinterAddr 9.152.120.241
NSinterAddr 9.152.64.172
NSportAddr 53
ResolveVia UDP
ResolverTimeout 5

```

TCP/IP profile

```

;-----
;Dynamic VIPA Definition
VIPADYNAMIC
VIPARANGE DEFINE MOVEABLE DISRUPTIVE 255.255.255.240 10.101.5.192
VIPARANGE DEFINE MOVEABLE DISRUPTIVE 255.255.255.240 10.101.6.192
VIPARANGE DEFINE MOVEABLE DISRUPTIVE 255.255.255.252 9.152.20.160
VIPARANGE DEFINE MOVEABLE NONDISRUPTIVE 255.255.255.248 9.152.20.176
ENDVIPADYNAMIC
;-----
;Source IP for NFS Server VIPA (required for NFSv3 clients)
SRCIP
JOBNAME MVSNFSHA 10.101.5.193
ENDSRCIP
;-----
;Settings for the TCP layer
TCPCONFIG
    INT 10           ; KeepAlive Interval 10 Min. (Default 120)
    SENDG TRUE      ; KeepAlive Probes with 1 Byte 'garbage'
    UNRESTRICTL     ; LOW PORTS 1-1023 not reserved (Default)
    TCPRCVB 65536   ; TCPReceiveBuffer in Bytes (z/OS 2.1 or 2.2: Default 64k)
    TCPSENDB 65536 ; TCPSendBuffer in Bytes (z/OS 2.1 or 2.2: Default 64k)

```

```

;-----
; Allow TCP/IP to determine the max MTU size on PATH
IPCONFIG
PATHMTUDISCOVERY
;-----
; Allow HiperSockets to use multiple write support and exploit zIIPs
GLOBALCONF
IQDMULTIWRITE
ZIIP IQDIOMULTIWRITE
;-----
;Device and Link definitions
DEVICE VLINK1 VIRTUAL 0
LINK VLINK1 VIRTUAL 0 VLINK1
;
;- z/OS 2.1 or 2.2 Use INTERFACE statement for HiperSockets - -
INTERFACE HIPERE4
DEFINE IPAQIDIO
IPADDR 10.101.4.68/26
SOURCEVIPAINTERFACE VLINK1
CHPID E4
;
INTERFACE COH2L1
DEFINE IPAQENET
IPADDR 9.152.20.136/22
PORTNAME OSAPORT
MTU 1492
NONROUTER
;
INTERFACE COHVS2
DEFINE IPAQENET
IPADDR 10.101.4.131/26
PORTNAME COHGE1
SOURCEVIPAINTERFACE VLINK1
MTU 8992
OLM NOISOLATE
PRIROUTER
;
;-----
; HOME internet (IP) addresses of each link in the host
HOME
10.101.4.210 VLINK1 ; VIPA
;
PRIMARYINTERFACE VLINK1
;-----
;IP routing information
BEGINROUTES
; IPV4
; NETWORK MASK FIRST HOP LINK PCKTSZ
ROUTE 9.152.20.0/22 = COH1L1 MTU 1492
ROUTE DEFAULT 9.152.20.1 COH1L1 MTU 1492
ENDROUTES
;-----
; Start all the defined devices.
START COH2L1
START COHVS2
START HIPERE4 ; HIPERSOCKET
;-----
;Reserve ports for the following servers
PORT
111 TCP RPCBIND1 ; RPC SERVICE
111 UDP RPCBIND1 ; RPC SERVICE
1223 TCP OMVS ; OE TELNET SERVER
1389 TCP BBOLDAP ; PORT NUMBER FOR BBOLDAP
8803 TCP RMFDDS01 BIND 9.152.20.177 ; RMF DDS VIPA FOR SAP SYSPLEX MONITORING
GLOBALTCPDATA(/etc/resolv.conf)
GLOBALIPNODES(/etc/hosts)
COMMONSEARCH

```

```

;-----
; Flush the ARP tables every 5 minutes
ARPPAGE 5
;-----
;Specify maximum length for the connection request queue created by
; the socket call listen().
SOMAXCONN 1024 (z/OS 2.1 or 2.2: Default 1024)

```

OMPROUTE start procedure

```

//OMPROUTE PROC
//OMPROUTE EXEC PGM=OMPROUTE,REGION=4096K,TIME=NOLIMIT,
// PARM=('POSIX(ON)',
//      'ENVAR("_CEE_ENVFILE=DD:STDENV")/')
//STDENV DD PATH='/etc/omproute.stdenv',
//      PATHOPTS=(ORDONLY)
//SYSPRINT DD PATH='/tmp/omproute.stdout',
//      PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//      PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//SYSOUT DD PATH='/tmp/omproute.stderr',
//      PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//      PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//
//SYSERR DD SYSOUT=*
//CEEDUMP DD SYSOUT=*

```

OMPROUTE environment variables

```

RESOLVER_CONFIG=/etc/resolv.conf
OMPROUTE_FILE=/etc/omproute.conf
OMPROUTE_DEBUG_FILE=/tmp/omproute.debug

```

Define the OMPROUTE procedure to RACF. At a TSO command prompt, enter the following commands:

```

rdefine started omproute.* stdata(user(stcuser) group(stcgroup))
setr raclist(started) refresh

```

OSPF routing parameters

The important things to note about the routing definitions are:

- The MTU must be the same for communication by all OSPF daemons on the same Ethernet segment.
- 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.

OMPROUTE configuration in /etc/omproute.conf:

```

Area
    Area_Number=0.0.0.0
    Stub_Area=NO
    Authentication_Type=None
    Import_Summaries=YES
;
ROUTERID=10.101.4.131;
;
GLOBAL_OPTIONS ignore_undefined_interfaces=yes
;
Interface
    IP_Address=9.152.20.136
    Name=COH2L1
    Subnet_Mask=255.255.252.0
    MTU=1492

```

```

;
OSPF_Interface
  IP_Address=10.101.4.131
  Name=COHVS2
  Subnet_Mask=255.255.255.192
  Attaches_To_Area=0.0.0.0
  MTU=8992
  Cost0=15
  Router_Priority=11
  Parallel_OSPF=Primary
  Hello_Interval=10
  Dead_Router_Interval=40
  Retransmission_Interval=5
  DB_Exchange_Interval=120
;
OSPF_Interface
  IP_Address=10.101.4.67
  Name=HIPERE4
  Subnet_Mask=255.255.255.192
  Attaches_To_Area=0.0.0.0
  MTU=8192
  Cost0=10
  Router_Priority=11
  Parallel_OSPF=Primary
  Hello_Interval=10
  Dead_Router_Interval=40
  Retransmission_Interval=5
  DB_Exchange_Interval=120
;
;OSPF_Interface statement for a virtual (VIPA) interface
;
OSPF_Interface
  IP_Address=10.101.4.211
  Name=VLINK1
  Subnet_Mask=255.255.255.255
  Attaches_To_Area=0.0.0.0
  Cost0=1
  MTU=8992
;
;OSPF_Interface statement for VIPARANGE
;
OSPF_Interface
  IP_Address=10.101.5.192
  Subnet_Mask=255.255.255.240
  Name=VIPL1
  Cost0=1
  MTU=8992

```

Linux on z Systems network settings

This information describes various aspects of the required network setup of the Linux on z Systems network.

Quagga setup - OSPF

/etc/quagga/ospfd.conf

```

hostname ihlscoh2
password quagga
enable password quagga
!
interface dummy0
  ip ospf cost 1
!
interface eth1
  ip ospf cost 15
!

```

```

interface hsi0
  ip ospf cost 10
!
router ospf
  ospf router-id 9.152.20.158
  network 10.101.4.215/32 area 0
  network 10.101.4.128/26 area 0
  network 10.101.4.64/26 area 0
!
line vty
!
log file /var/log/quagga/ospfd.log

```

Zebra setup - Zebra

/etc/quagga/zebra.conf

```

hostname ihlsco2
password quagga
enable password quagga
ip route 0.0.0.0/0 9.152.20.1
route-map vipal permit 10
  match ip address prefix-list DEST
  set src 10.101.4.215
  continue
route-map vipal permit 20
ip protocol ospf route-map vipal
ip prefix-list DEST permit 10.0.0.0/8 le 32
log file /var/log/quagga/quagga.log

```

AIX OSPF definitions for the gated daemon

Read this information to find AIX OSPF definitions for the gated daemon.

/etc/gated.conf:

```

interfaces {
interface all passive;
};
#
routerid 10.101.4.240;
#
traceoptions "/tmp/gated.log" size 4m files 10 general;
# traceoptions none;
#
rip off;
egp off;
bgp off;
ripng off;
hello off;
isis off;
redirect off;
snmp off;
#
ospf yes {
backbone {
networks {
10.101.4.240 mask 255.255.255.255;
10.101.4.128 mask 255.255.255.192;
};
interface 10.101.4.240 cost 1 {
enable;
# hellointerval 10;
# routerdeadinterval 40;
# retransmitinterval 5;
# priority 0;
};
interface 10.101.4.140 cost 15 {

```

```

enable;
# hellointerval 10;
# routerdeadinterval 40;
# retransmitinterval 5;
# priority 20;
};
};
};
static {
default
gateway 9.152.20.1 # BB-Lan
preference 160 # administrative cost worse than 150 = OSPF
retain; # do not remove at graceful termination
};
;import proto ospfase { } ;
;export proto ospfase { } ;

```

Static VIPA definitions required for SUSE

Read this information to find static VIPA definitions required for SUSE.

To ensure that the dummy kernel module is loaded at boot time, one solution is to add a line similar to the following into `/etc/sysconfig/kernel`:

```
MODULES_LOADED_ON_BOOT="vmcp dummy"
```

For SLES 11 set the dummy0 interface definition:

```

BOOTPROTO="static"
UNIQUE=""
STARTMODE="onboot"
MTU="8992"
IPADDR="10.101.4.216"
NETMASK="255.255.255.255"
NETWORK="10.101.4.216"
BROADCAST="0.0.0.0"

```

The shown new dummy0 interface can be started using `ifup`, and it is automatically started at boot time.

For any local OSA interfaces you need to add the following statement to their `ifcfg-*` file:

```
POST_UP_SCRIPT="setvipa"
```

This statement runs a script named `setvipa` when `ifup` processes the local interfaces `ifcfg-*` file.

Here is a SLES 11 example file `ifcfg-qeth-bus-ccw-0.0.0600` for `eth1`:

```

BOOTPROTO="static"
UNIQUE=""
STARTMODE="onboot"
MTU="8992"
IPADDR="10.101.4.137"
NETMASK="255.255.255.192"
NETWORK="10.101.4.128"
BROADCAST="10.101.4.191"
_nm_name='qeth-bus-ccw-0.0.0600'
POST_UP_SCRIPT="setvipa"

```

Note: With SLES 12, the *wicked* framework has been introduced. If you are running the *wicked* package with minimum version 0.6.18 and release 16.1, then the entry must look like:

```
POST_UP_SCRIPT="compat:suse:setvipa"
```

Download the current *wicked* version, for example, from the following website:
<https://www.suse.com/>

The following `setvipa` script should be placed into `/etc/sysconfig/network/scripts/` as `setvipa`:

```
#!/bin/bash
#
# This script is called via ifup when it
# processes an ifcfg--* script that contains:
#   POST_UP_SCRIPT="setvipa"
#
# First obtain interface name
INT=$2
#
# Now read in the dummy0 VIPA details
. /etc/sysconfig/network/ifcfg-dummy0
#
# Copy the IP address for dummy0
VIPA=${IPADDR}
#
# Set the VIPA address into the OSA interface
/sbin/qethconf vipa add ${VIPA} ${INT}
```

Using the command `ifup eth1` gives output similar to the following:

```
ifup eth1
    eth1
        eth1 configuration: qeth-bus-ccw-0.0.0600
qethconf: Added 10.101.4.216 to sysfs entry /sys/class/net/eth1/device/vipa/add4.
qethconf: For verification please use "qethconf vipa list"
```

NFS client automount samples

The subtopics contained in this information unit include automount samples for NFS clients on Linux and AIX.

Linux sample with NFSv3 mounts

The Linux automount daemon is controlled via `/etc/init.d/autofs`, and can be started, stopped, or restarted via the service command as described in this topic.

The `autofs` service can be configured to start at boot time by issuing the command:
`chkconfig -a autofs`

The service can be manually started using the following command:

```
service autofs start
```

or

```
rcautofs status/stop/start/restart
```

Here is the `auto.master` configuration file:

/etc/auto.master

```
# Sample auto.master file
/sapmnt/HA1 auto.ha1.sapmnt
```

Here is the referenced file for `sapmnt auto.ha1.sapmnt`:

/etc/auto.ha1.sapmnt

```

| exe      -nfsvers=3,rw,hard,intr,rsz=8192,wsz=8192 sapnfsv:/hfs/sapmnt/HA1/exe,binary,mvsmnt,rdrverf
| profile -nfsvers=3,rw,hard,intr,rsz=8192,wsz=8192 sapnfsv:/hfs/sapmnt/HA1/profile,TEXT,mvsmnt,rdrverf,
|         cln_ccsid(819),srv_ccsid(1047)
| global  -nfsvers=3,rw,hard,intr,rsz=8192,wsz=8192 sapnfsv:/hfs/sapmnt/HA1/global,TEXT,mvsmnt,rdrverf,
|         cln_ccsid(819),srv_ccsid(1047)
| trans   -nfsvers=3,rw,hard,intr,rsz=8192,wsz=8192 sapnfsv:/hfs/sap/transA11/HA1trans,TEXT,mvsmnt,rdrverf,
|         cln_ccsid(819),srv_ccsid(1047)

```

AIX sample with NFSv4 mounts

Find a list of commands to start, stop, or verify the AIX automount daemon.

The AIX automount daemon can be started, stopped, or verified via command:

```
startsrc/stopsrc/lssrc -s automountd
```

Here is the auto_master config file:

```

/etc/auto_master
# Sample auto_master file
/sapmnt/HA2 auto.ha2.sapmnt

```

Here is the referenced file for sapmnt auto.ha2.sapmnt:

```

/etc/auto.ha2.sapmnt
exe      -rw,vers=4,hard,intr,sec=sys sapnfsv:/HFS/sapmnt/HA2/AIX/exe,binary,mvsmnt,rdrverf
global  -rw,vers=4,hard,intr,sec=sys sapnfsv:/HFS/sapmnt/HA2/global,text,mvsmnt,rdrverf,
         cln_ccsid(819),srv_ccsid(1047)
profile -rw,vers=4,hard,intr,sec=sys sapnfsv:/HFS/sapmnt/HA2/profile,text,mvsmnt,rdrverf,
         cln_ccsid(819),srv_ccsid(1047)
trans   -rw,vers=4,hard,intr,sec=sys sapnfsv:/HFS/sap/transA11/HA2trans,text,mvsmnt,rdrverf,
         cln_ccsid(819),srv_ccsid(1047)

```

Migration considerations

This information presents various subtopics that provide guidance on NFSv4 migration and SAP System Automation policy migrations. It also presents helpful hints and tips on Linux for SLES 11 SP2.

Note: NFSv4 client for Linux on z Systems or Linux on System x is not recommended in a high availability setup with a z/OS NFS server movable between LPARs. Read *SAP Note 2099374* for further information.

The following topics are discussed:

- “NFSv4 migration hints and tips”
- “SAP System Automation policy migrations” on page 311
- “Linux migration hints” on page 313

NFSv4 migration hints and tips

This information provides easy to follow guidelines to help you set up an NFS client and server configuration for NFSv4. The configuration covers an environment with NFS Server z/OS and NFSv4 client AIX, it does not cover a configuration with Kerberos security.

The instructions in this topic are intended for, but not restricted to, an NFSv4 implementation in a *business continuity solution* environment. The instructions can be useful for any NFSv4 client-server configuration in the context of z/OS and AIX.

Note: In a High Availability environment it is currently not recommended to use the NFSv4 client on Linux because the current NFSv4 client on Linux is not able to recover from a failover of the z/OS NFS Server to a different z/OS LPAR.

The following subtopics are discussed:

- “Introduction to NFSv4 features”
- “NFSv4 features that differentiate it from NFSv3”
- “A sample environment” on page 304
- “Configuring and verifying the NFS client and server for NFSv4” on page 305
- “Useful commands for testing and debugging” on page 309
- “File samples for z/OS and z/OS UNIX” on page 309

Related information

- *z/OS Communications Server: IP Configuration Guide*
- *z/OS Communications Server: IP Configuration Reference*
- *AIX 7.1 information - Network File System*

Introduction to NFSv4 features

The z/OS Network File System (NFS) is a product that provides IBM system-managed storage to the network environment. It lets you optimize efficiency in a distributed network while still capitalizing on the capacity, security, and integrity of z/OS.

The following is a brief overview of NFSv4 features:

- Lock and mount protocols are now integrated into the NFS protocol
- Stateful operations (operate on client and server outages)
- Built-in security with RPCSEC_GSS (based on GSS-API) Requires either:
 - Kerberos 5
 - LIPKEY (based on SPKM-3)
 - SPKM-3
- Makes extensive use of client-side caching
- Supports replication and migration
- IETF standard protocol RFC 3530
- Supports Unix-like clients as well as Windows clients
- Supports ACLs

NFSv4 replaces older NFS versions.

NFSv4 features that differentiate it from NFSv3

The subsections of this topic describe some major features of NFSv4, which differentiate NFSv4 from NFSv3.

The following topics are contained:

- “Pseudo-root file system concept” on page 304
- “Accessing symbolic links on z/OS with NFSv4” on page 304
- “NFSv4 and lock manager” on page 304

Pseudo-root file system concept

A specific difference between NFSv3 and NFSv4 is the NFSv4 *pseudo-root* file system concept. There are different implementations in z/OS NFS Server and Linux NFS Server. The following example reflects the z/OS implementation, which is based on NFSv4 RFC-3550.

Server pseudo file system

NFSv4 servers avoid name space inconsistency by presenting all the exports within the framework of a single server name space. An NFSv4 client uses LOOKUP and READDIR operations to browse seamlessly from one export to another. Portions of the server name space that are not exported are bridged via a *pseudo file system* that provides a view of exported directories only. A pseudo file system has a unique ID and behaves like a normal, read only file system.

Based on the construction of the server's name space, it is possible that multiple pseudo file systems may exist. For example:

File system	Meaning
/a	pseudo file system
/a/b	real file system (need to be exported)
/a/b/c	pseudo file system
/a/b/c/d	real file system (need to be exported)

Each of the pseudo file systems are considered separate entities and therefore have a unique ID.

Accessing symbolic links on z/OS with NFSv4

When you use an exports file with Security(EXPORTS) or Security(SAFEXP), both the initial path containing the symbolic link and the target path must be exported. If the symbolic link is not exported, the initial mount processing fails. After the symbolic link is discovered, the NFS client starts over with the mount emulation (lookup) processing using the target path name. If that path is not exported, then that mount processing fails. See APAR OA33155 for a detailed explanation.

NFSv4 and lock manager

With NFSv4, the lock manager is part of the NFSv4 protocol implementation. There is no extra lock manager as with earlier versions of NFS such as with NFSv3. This has an impact on NFS attribute settings and SOURCEVIPA settings. See "Configuring the NFSv4 server on z/OS" on page 305.

A sample environment

The described sample environment includes one SAP system HA2.

HA2 is an SAP 7.10 PI installation with three application servers on AIX, SAP Central Services on z/OS, and a DB2 data sharing group on z/OS.

The SAP cluster uses shared file systems that are managed by z/OS NFS Server.

The SAP cluster itself is managed by Tivoli System Automation products.

The SAP file systems on z/OS

The following is an example definition of an SAP file system on the HA2 sample SAP system.

```
/hfs/sapmnt/HA2/exe  
/hfs/sapmnt/HA2/global  
/hfs/sapmnt/HA2/profile  
/hfs/sap/transA11/HA2trans
```

Configuring and verifying the NFS client and server for NFSv4

This information describes the required steps to configure and verify the NFS client on z/OS and the NFS server on AIX for NFSv4.

You need to process the following steps to configure and verify the NFS client on z/OS and the NFS server on AIX for NFSv4:

1. “Configuring the NFSv4 server on z/OS”
2. “Configuring the NFSv4 client on AIX” on page 306
3. “Verification scenarios” on page 307

Configuring the NFSv4 server on z/OS:

This topic describes how to configure the NFS server on z/OS.

Several data sets and files require your attention. See “File samples for z/OS and z/OS UNIX” on page 309 for examples.

z/OS hlq.PARMLIB(BPXPRMxx) must include the initial mount statements on z/OS and a link to the user alias table in z/OS UNIX.

hlq.PARMLIB(<dynamic vipa configuration>) must include a VIPA range definition that includes the NFS server VIPA IP address. With NFSv4 you do not need to define SRCIP for the NFS VIPA. In a NFSv4-only configuration you could remove the SRCIP statements for the NFS VIPA. In a mixed configuration with NFSv3 and NFSv4 you should keep the SRCIP statements for the NFS VIPA.

hlq.PARMLIB(<NFS exports>) must include the file system export statements. Allow root access for the NFS client to dedicated clients only. The use of the access and root options allows R/W root access to the listed clients and prohibits any access from any other clients. The following sample entries grant R/W root access to the three clients. Any access from any other clients is denied:

```
/hfs/sapmnt -access=10.101.4.214<root>|\  
10.101.4.215<root>|\  
10.101.4.216<root>
```

hlq.PARMLIB(<NFS attributes>):

- **id2name attribute**

z/OS 2.2 introduces a new NFS server site attribute **id2name**. We recommend to explicitly use `id2name(cache)` in order to enhance NFSv4 performance by utilizing UID/GID caching and eliminating calls to RACF. This attribute affects only NFSv4. The default is `id2name(callsaf)`.

- **nlm attribute**

In an NFSv4-only configuration you should set this attribute to `nonlm`. In a mixed configuration with NFSv3 and NFSv4 you should set this attribute to `nlm`.

- **Remount attribute**

This attribute is required only in an NFSv3 environment or in a mixed environment with NFSv3 and NFSv4 mounts to the same NFS server. The REMOUNT attribute should be removed in pure NFSv4 environments. When removed it defaults to NOREMOUNT.

When used, the REMOUNT attribute enables the NFS server to process NFS requests after the NFS server is restarted even though the HFS file system was remounted with a new HFS file system number (device number) after its last usage. Use of the REMOUNT attribute causes the NFS server to automatically access a remounted HFS file system even though it may have been changed before remounting. Any active client mounts are reestablished.

- **nfsv4domain(NFSv4_default_domain) attribute**

This attribute should be appropriately set. NFSv4_default_domain specifies the "pseudo" NFSv4 domain for the NFSv4 name mapping. The "pseudo" NFSv4 domain allows various NFSv4 Clients from various network domains to seamlessly access the server provided that these NFSv4 clients are also configured with the same domain.

If the NFSV4DOMAIN attribute is not used, the server uses the system-defined domain. The participating NFSv4 client domains must match one of the server's network domain for proper NFSv4 name mapping.

NFSv4domain attribute support is available with APAR OA30333, PTF UA52468. For further details refer to the latest z/OS NFS Guide and Reference. There is no client support yet for the NFSV4DOMAIN server attribute so for now the NFSV4DOMAIN is not exploited in this installation.

Note: Name resolution is not supported through any global name server such as LDAP.

z/OS UNIX

/etc/ualiastable

NFSv4 since z/OS 1.10 requires user alias mapping. The same owner and group names are to be defined on both the server and client. The owner and group names must be defined to RACF with appropriate *uid* and *gid* values on z/OS. **/etc/resolv.conf**

Must include DOMAINORIGIN set to a specific domain name.

Note: The domain name must be equal on participating NFSv4 server and clients.

Configuring the NFSv4 client on AIX:

This topic describes how to configure the NFSv4 client on AIX.

Prerequisites: For AIX 6.1 an operating system level of 6100-05-01-1016 or higher.

1. Configure the domain name:

- a. Configure the domain name to the same name as defined in z/OS UNIX `/etc/resolv.conf` or as defined with z/OS NFS Server attribute `NFSV4DOMAIN`.
 - b. Execute `smitty chnfsdom`.
 - c. Change NFS Local Domain to `<domain name of z/OS UNIX resolv.conf>`.
 - d. Verify the domain name setting.
 - e. Execute `chnfsdom`.
2. **Configure the automounter service:**

Modify `auto_master` and create an `auto.ha2.sapmnt` to allow the automounter to mount the SAP file systems on demand with NFSv4 protocol.

The mount statements must include `vers=4` and mount options `mvsmnt` and `rdrverf`.

Note: The `/etc/auto_master` file is only read when the automount command is initially executed. Changes to it do not take effect until the automount command is run again.

3. **Configure NFS client service**
- a. Start the NFS services that are required for NFSv4.
 - b. Execute `startsrc -s nfsrgyd`
 - c. Verify the NFS services.
 - d. Execute `lssrc -g nfs`

The services `nfsrgyd` and `nfsd` show active in the **Status** column.

Verification scenarios:

This information unit describes verification scenarios for failover, data integrity, and SAP integrity.

Failover

1. Move NFS Server:
 - a. Logon to SAP GUI and execute SAP workload with transaction `sgen`
 - b. Move the NFS from System A to System B (use SA z/OS `INGMOVE`)

The NFS server moves to a different system.

The NFS mounted SAP file systems are available before and after the move.

Transaction `sgen` continuously runs without error messages.

No errors are reported on the NFS client side when executing the list command: `ls -a1R /sapmnt/HA2/* > /dev/null`

2. Shut down the system with NFS Server:
 - a. Log on to SAP GUI and execute SAP workload with transaction `sgen`
 - b. Shut down the system where the NFS server actually runs.

The NFS server moves to a different system.

The NFS mounted SAP file systems are available before and after the move.

Transaction `sgen` continuously runs without error messages.

No errors are reported on the NFS client side when executing the list command: `ls -a1R /sapmnt/HA2/* > /dev/null`

Data integrity

1. On a NFS client copy large the NFS mounted files during a NFS server move.

2. Simultaneously initiate the NFS Server move and a the copy command. Use SA z/OS INGMOVE to move z/OS resources. This NFS client message is displayed:

```
$ cp -r <large directory tree> testnfsv4
NFS server <NFS server hostname> not responding still trying
NFS server <NFS server hostname> ok
$
```

3. Compare the source and destination directory tree:

```
diff -b -h <large directory tree> testnfsv4
```

<large directory tree> and testnfsv4 must be equal.

SAP integrity

1. Initiate an SAP application server start during an NFS server move. Use System Automation for Multiplatforms to manage SAP resources on AIX and use SA z/OS INGMOVE to move z/OS resources.

2. To stop your SAP Application Server, run:

```
chrg -o offline <SAP Application Server resource name>
```

Wait until the application is offline.

3. Simultaneously start the SAP application and move the NFS server.

4. To start your SAP Application Server and initiate an NFS Server move, run:

```
chrg -o online <SAP Application Server resource name>
```

Wait until SAP application is online

5. Log on to SAP and run SAP System Check transaction sick and Diagnostics Dictionary->Database Consistency Check transaction db2, which gives you a quick verification of your SAP system

Tool supporting test and verification of NFS HA setup

Read this information about available scripts that support automated testing and verification of your NFS and file systems setup.

The NFS verification steps described in “Configuring and verifying the NFS client and server for NFSv4” on page 305 are specifically designed for manual processing and verification. An automated test and verification supported by tools would be desirable when you plan to frequently run a verification of your NFS and file systems setup.

Therefore, for verification of the used test environment, a set of scripts is available, which perform these tasks automatically. The scripts log-in on the NFS server and the NFS client hosts. They copy, move, and remove the NFS mounted copies of real SAP file system directory structures, locally and remotely. The scripts also move the NFS server from one system to the next and verify and compare the NFS file system content before and after the move.

Disclaimer: These scripts can greatly serve as a base for individual extension based on your local requirements. They are not intended and not developed to be the overall solution for each and all environments. Individual adaptation is required. These scripts together with a brief documentation are available from the *IBM developerWorks: SAP on IBM z Systems Community*. Click on Business Continuity Utilities and look for *NFS setup verification test* for additional information.

Useful commands for testing and debugging

Refer to this information if you are looking for commands for debugging and verification on z/OS or AIX.

Table 42. Debugging and test commands

Operating system	Command
z/OS	<ul style="list-style-type: none">• /D OMVS• /F DFSZFS,QUERY,SETTINGS• NETSTAT ROUTE
AIX	<ul style="list-style-type: none">• startsrc/stopsrc/lssrc -g nfs• startsrc/stopsrc/lssrc -s nfsrgyd• startsrc/stopsrc/lssrc -s automountd• rpcinfo -p• showmount -e <virtual NFS server hostname>

Test:

Check whether the NFSv4 mounted file systems are available.

AIX:

```
ls -a1R /sapmnt/HA2/*
```

This command lists all files recursively in all directories and subdirectories of mount point /sapmnt/HA2.

File samples for z/OS and z/OS UNIX

Read this information if you look for file samples for the indicated platforms. The file samples mainly include statements that are relevant to this NFSv4 implementation. The files in your environment may include other statements.

hlq.PARMLIB(<dynamic vipa configuration>)

```
VIPADYNAMIC
VIPARANGE DEFINE MOVEABLE DISRUPTIVE 255.255.255.240 <NFS server vipa IP address>
ENDVIPADYNAMIC
```

hlq.PARMLIB(<NFS exports>)

```
#
|
| # sapmnt file system export for AIX cluster
| /hfs/sapmnt -access=<hostname1><root>|\
|             <hostname2><root>|\
|             <hostname3><root>
|
| # trans file system export for AIX cluster
| /hfs/sap/transAll -access=<hostname1><root>|\
|                  <hostname2><root>|\
|                  <hostname3><root>
```

hlq.PARMLIB(<NFS attributes>)

```
space(100,10), blks
norlse
recfm(vb), blksize(0), lrecl(8196)
dsorg(ps)
dsntype(pds)
dir(27)
keys(64,0)
recordsize(512,4K)
```

```

nonspanned
shareoptions(1,3)
attrtimeout(120), readtimeout(90), writetimeout(30)
binary
/* Note: For a Windows application server, you must use the processing attribute */
/* 'text'. It is also recommended to use the 'tag' processing attribute. */
/* If you used the processing attribute 'binary' setting before, then you should change */
/* the client mounts under AIX or Linux for the executable directories to explicitly */
/* use the 'binary' option, if not already used. */
/* AIX: mount -o rw,vers=4,hard,intr,sec=sys '<virt. hostname of NFS server>: */
/* /HFS/sapmnt/<sid>/exe,binary,mvsmnt,rdrverf' /sapmnt/<sid>/exe */
/* Linux: mount -t nfs -o rw,vers=3,hard,intr,sec=sys '<virt. hostname of NFS server>: */
/* /HFS/sapmnt/<sid>/exe,binary,rdrverf' /sapmnt/<sid>/exe */
LF
blankstrip
mapleaddot
maplower
retrieve
nofastfilesize
setownerroot
executebitoff
xlat(oemvs311)
nofileextmap
security(exports)
nopcnsd
leadswitch
mintimeout(1)
nomaxtimeout
logout(20000000)
nfstasks(16,9)
restimeout(720,0)
cachewindow(112)
hfs
logicalcache(16M)
bufhigh(128M)
percentsteal(20)
readaheadmax(128K)
maxrdfsleft(32)
smf(none)
sfmax(0)
remount # for more information refer to the description
        # of hlq.PARMLIB(<NFS attributes>)
        # in configuring the NFS server on z/OS
id2name(cache) # Requires a z/OS level of z/OS 2.2 or higher
                # NFS performance improvement
                # Affects only NFS version 4

nonlm

```

hlq.PARMLIB(BPXPRMxx)

```

MOUNT FILESYSTEM('<ZFS-FILE-SYSTEM-SAPMNT>')
      MOUNTPOINT('/sapmnt')
      TYPE(ZFS)
      MODE(RDWR)
      AUTOMOVE
MOUNT FILESYSTEM('<ZFS-FILE-SYSTEM-TRANS>')
      MOUNTPOINT('/usr/sap/transAll')
      TYPE(ZFS)
      MODE(RDWR)
      AUTOMOVE

```

/etc/ualiastable

```

:userids
HA2ADM      ha2adm
SAPADM      sapadm
<root user> root

```



```
TCPIP      root
:groups
SAPINST    sapinst
SAPSYS     sapsys
```

/etc/resolv.conf

For a file sample of /etc/resolv.conf see “z/OS COH1 – Unix System Services network settings” on page 295.

SAP System Automation policy migrations

This topic contains information about policy migration in the areas of SAP Central Services and SAP Application Servers. It also contains a subtopic about the migration from PORTMAP to RPCBIND.

The following topics are discussed:

- “SAP Central Services policy migration”
- “From PORTMAP to RPCBIND”

SAP Central Services policy migration

This topic describes what you need to do to move from an existing *SAP policy for SAP Central Services of an existing SAP system to the *SAPSRV *add-on policy*.

There is no migration path for an existing SAP policy, based on the old *SAP *add-on policy*, to the policy structure that comes with the *SAPSRV *add-on policy* in SA 3.4. This is because the *SAPSRV *add-on policy* is based on the sapstartsrv infrastructure from SAP, and the old *SAP policy is not. However, it is possible to have different SAP systems that are automated via either *SAP- or *SAPSRV-based policies in parallel.

If you want to exploit the *SAPSRV *add-on policy* for an already automated SAP system, the recommended approach is:

1. ensure that all prerequisites for the *SAPSRV *add-on policy* are fulfilled
2. Create the policy for the SAP system from scratch using the new HA wizard.
3. Remove the existing policy definitions for that SAP system in your production PDB.

This means that you must delete the APGs, APLs, and classes in the customization dialog, which start with SAP<SID> (<SID> is the SAP System ID) as *Entry Name*, and which are not proxy resources for remote Application Servers. For example, for an SAP system with SID HA2, delete the APGs and APLs starting with SAPHA2, and delete the classes starting with C_SAP_HA2, but do not delete potentially existing proxy resources for remote the Application Server. Do not delete the common, unchanged class C_SAP_USS.

4. Import the newly generated policy for your SAP system into your production PDB.

Note: This procedure means several downtimes for the SAP system which you want to change:

- downtime for implementing the prerequisites
- downtimes for tests.

From PORTMAP to RPCBIND

This task describes how to move safely from the PORTMAP to the RPCBIND service.

About this task

The following steps can help you to move safely from PORTMAP to RPCBIND service:

Procedure

1. Create a PROCLIB member RPCBIND similar to PORTMAP

```
//*****  
//RPCBIND PROC  
//*  
//* TCP/IP FOR MVS  
//* SMP/E DISTRIBUTION NAME: EZARBBND  
//* FUNCTION: UNIX SYSTEM SERVICES RPCBIND SERVER MAIN PROCESS  
//*  
//RPCBIND EXEC PGM=RPCBIND,REGION=4096K,TIME=1440  
//*  
//STDOUT DD SYSOUT=*  
//STDERR DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
// PEND
```

2. (Optional) Change TCP/IP profile to autolog RPCBIND instead of PORTMAP

Note: Skip this step in case you want to manage RPCBIND as a SA z/OS resource.

```
;AUTOLOG  
    RPCBIND ; RPCBIND is required for NFS  
;ENDAUTOLOG
```

3. (Optional) Change TCP/IP profile to reserve port 111 tcp and udp for RPCBIND instead of PORTMAP

Note: This step is required if you have port reservation for portmapper. Edit the TCP/IP profile and change from PORTMAP to RPCBIND. Skip this step if you do not have port reservation for portmapper and if you do not intend to have port reservation for RPCBIND.

```
PORT  
    111 TCP RPCBIND ; RPCBIND SERVER  
    111 UDP RPCBIND ; RPCBIND SERVER
```

4. Update RACF For the definition of the started task user and further security settings for RPCBIND see the sample jobs that are provided in <hlq>.SEZAINST(EZARACF). Do a FIND on RPCBIND.

5. SA z/OS 3.3 update policy

a. Modify application RPCBIND:

- Add a *HasParent* relationship to TCPIP/APL/=
- Add a *ForceDown* relationship to TCPIP:
 - Relationship: *ForceDown*
 - Supporting resource: TCPIP/APL/=
 - Condition: *WhenObservedHardDown*

b. Modify application OMPROUTE:

- Modify the *ForceDown* relationship to TCPIP:
 - Relationship: *ForceDown*
 - Supporting resource: TCPIP/APL/=
 - Condition: *WhenObservedAssumedDownOrStopping*

Linux migration hints

This topic discusses some valuable aspects while migrating on Linux platforms.

Tips for migrating to SLES 11 SP2

Observe the described considerations when migrating to SLES 11 SP2.

SLES 11 SP2 introduces a new Linux kernel version 3. Before you upgrade to SLES 11 SP2, you might need to update your 7.20 kernel level. See *SAP Note 1310037: SUSE LINUX Enterprise Server 11: Installation notes* and *SAP Note 1629558: Linux 3.x kernel*.

SLES 11 SP2 also requires to upgrade your SA MP software on SLES 11 SP2 to version 3.2.2.2. or a later version.

Tips for migrating from SLES 11 to SLES 12

Observe the described considerations when migrating from Linux SLES 11 to SLES 12.

If you want to use SAP software on SLES 12, apply the information that is provided in *SAP Note 1984787*. This note discusses general instructions for installing SLES 12 for all versions. Furthermore, it provides:

- additional installation instructions for all platforms
- information about supported hardware platforms
- upgrading information from SLES 11 to SLES 12
- known issues and workarounds on SLES 12

Note: With SLES 12, the *wicked* framework has been introduced. Run the *wicked* package with minimum version 0.6.18 and release 16.1 to circumvent the problem that **POST_UP_SCRIPT** option in `ifcfg-eth<x>` does not work.

Download the current *wicked* version, for example, from the following website:
<https://www.suse.com/>

In addition to the instructions from *SAP Note 1984787*, implement the updates, fixes, and recommendations for the SLES 11 to SLES 12 migration that are described in the following sections:

- “NFS client adaption”
- “systemd adaption” on page 314
- “Shutdown problem” on page 314

NFS client adaption:

NFS client mounts defined under SLES 11 to a highly available NFS server under z/OS might not work anymore under SLES 12. This is because the default mount behavior changed with SLES 12. The default mount type of the NFS client under SLES 11 is `nfs v3` (version 3) as with SLES 12 it is `nfs v4` (version 4).

Change all mount points for the z/OS NFS server as shown for the global directory in an automounter setup:

```
global -nfsvers=3,rw,hard,intr,rsize=8192,wsiz=8192 sapnfsv:/hfs/sapmnt/HA3/global,  
TEXT,mvsmnt,cln_ccsid(819),srv_ccsid(1047)
```

It means that with SLES 12 you must use the **-nfsvers=3** option to be sure to operate on `nfs v3` mounts. The man page of `nfs` says:

nfsvers=*n*

is the NFS protocol version number used to contact the server's NFS service. If the server does not support the requested version, the mount request fails. If this option is not specified, the client negotiates a suitable version with the server, trying version 4 first, version 3 second, and version 2 last.

Refer to *SAP Note 2099374 (DB2-z/OS: NFSv4 client for Linux on IBM System z or x is not recommended in a High Availability setup with a z/OS NFS Server movable between LPARs)* to learn why version 4 is not recommended.

systemd adaption:

Read the contained information how to adapt systemd.

As a root user, perform the following to adapt the /etc/init.d/autofs script:

1. Adapt Start-dependency:

```
cp /usr/lib/systemd/system/autofs.service /etc/systemd/system
chmod 664 /etc/systemd/system/autofs.service
```

In /etc/systemd/system/autofs.service, change from:

```
After=network.target remote-fs.target nss-lookup.target nss-user-lookup.target
```

to:

```
After=network.target remote-fs.target nss-lookup.target
nss-user-lookup.target zebra.service ospfd.service
```

Then, check with command: **systemd-delta systemd/system**

2. Call the testNFS script before autofs:

Create and save the file /etc/systemd/system/beforeafs.service (664 permissions), which contains:

```
[Unit]
Before=autofs.service
[Service]
Type=oneshot
RemainAfterExit=true
ExecStart=/bin/sh -c '/root/testNFS'
```

systemd.service and systemd.unit for full syntax

```
[Install]
# target in which to start the service
WantedBy=multi-user.target
#WantedBy=graphical.target
```

Then, run:

```
systemctl daemon-reload
```

Check if the **beforeafs.service** is enabled and active (running) with the command:

```
systemctl status beforeafs.service
```

If **beforeafs.service** is not enabled or active, then run:

```
systemctl enable beforeafs.service
systemctl start beforeafs.service
```

Additionally, in the same way check if **autofs.service** is enabled and active.

Shutdown problem:

Read about how to react if shut down causes problems and takes a long time.

If shut down takes very long and on the z/VM console you see many messages like: A stop job is running for Automounts filesystems on ..., then change the line `+auto.master` in the `/etc/auto.master` file into a comment like: `#+auto.master`.

SA MP prerequisites when migrating to SLES 12

If you migrate to a Linux SLES 12 environment, and want to continue to use System Automation for Multiplatforms, then you need to follow the steps that are presented in this topic.

System Automation for Multiplatforms 4.1.0.1 introduces support for SLES 12. On SLES 12, SA MP runs as 64-bit application while it runs as 32-bit application in compatibility mode on SLES 11. A cluster that is created with a 64-bit SA MP cannot be used with a 32-bit SA MP, and also not the other way round. For this reason, SA MP offers no node-by-node migration when migrating from SLES 11 to SLES 12. On SLES 12, you must start with an initial installation and create a new cluster.

If your current setup is Linux SLES 11 SPx with SA MP 3.x or SA MP 4.x 32-bit, and you plan to migrate to SLES 12, then follow this approach:

1. Save your current SA MP policy by using the **sampolicy** command on one of the cluster systems. Write down customizations you applied to the cluster. See the SA MP documentation for details.

If you want to migrate your existing policy, you must save your current SA MP policy by using the **sampolicy** command on one of the cluster systems. Write down customizations you applied to the cluster. See the SA MP documentation for details. If you want to benefit from the enhanced capabilities of the SAP high availability policy feature of SA MP 4.1.0.2 or higher, you should generate a new SA MP policy for your SAP system (see “Using the SAP high availability policy feature to generate a policy” on page 186).

2. Shut down the cluster.
3. Remove the SA MP installation from your SLES 11 SPx systems. See the SA MP documentation for uninstall information. SA MP uninstallation leaves SA MP configuration files in `/etc/opt/IBM/tsamp/sam/cfg` on the system.
4. Migrate to SLES 12 on all systems
5. Install SA MP 4.1.0.1 64-bit on all systems. Usually, SA MP Fix Packs can be installed as update only. See the SA MP documentation for details on installing a Fix Pack as initial installation.
6. Re-create the cluster with customizations from step 1.
7. Restore the SA MP policy from step 1.

Tips for migrating from SLES 12 SP1 to SLES 12 SP2

Observe the described considerations when migrating from SLES 12 SP1 to SP2.

SLES 12 SP2 introduces a new Linux kernel version 4. SAP updates might be required. Read *SAP Note 2187639: Linux 4.x-Kernel* for additional information.

With SLES 12 SP2, the key exchange algorithms `diffie-hellman-group1-sha1`, `diffie-hellman-group-exchange-sha1` and the cipher `aes128-cbc` are no longer default. This may impact SSH connections to a SLES 12 SP2 server.

The algorithms currently offered by the SSH daemon can be displayed with command `sshd -T | grep kexalgorithms` and `sshd -T | grep ciphers`. Add any additionally required algorithms to the SLES12 SP2 `/etc/ssh/sshd_config` file.

Chapter 16. Bibliography

This topic provides references to a variety of information related to business continuity for SAP on z Systems.

Edition history of this publication and supported component versions

In a table, this topic presents information about which versions of the participating components have been supported in previous editions of *Business Continuity for SAP on IBM z Systems*.

Table 43. Edition history of *Business Continuity for SAP on IBM z Systems*

<i>Business Continuity for SAP on IBM z Systems</i>	Order number	SAP NetWeaver Releases	IBM Tivoli System Automation for z/OS	z/OS Releases
Edition 12/2014 [1]	SC33-8206-07	SAP NetWeaver 7.4 based on 7.4x kernel, and SAP NetWeaver 7.1, 7.2, and 7.3 based on the 7.20 or 7.21 Downward Compatible Kernel (7.20 / 7.21 DCK)	Version 3.5 and Version 3.4 with APAR level OA46166 or higher	z/OS V1R13 and higher supported releases
Edition 12/2015	SC33-8206-08	SAP NetWeaver 7.5 and 7.4 based on 7.4x Downward Compatible Kernel (DCK), and SAP NetWeaver 7.1, 7.2, and 7.3 based on 7.2x DCK levels.	Version 3.5 with APAR level OA48922 or higher	z/OS V2R1 and higher supported releases
Edition 12/2016 (this edition) [2]	SC33-8206-09	SAP NetWeaver 7.5x and 7.4 based on 7.4x kernels, and SAP NetWeaver 7.3 and lower based on 7.2x kernels.	Version 3.5 with APAR level OA51440 or higher	z/OS V2R1 and higher supported releases
[1]	This edition is the recommended reference for customers using SA z/OS 3.4 and customers using SA MP who want to implement an SA MP policy for SAP on the basis of a sample policy contained in previous versions of the zSAP_BusinessContinuity.zip file.			
[2]	This edition is the recommended reference for customers using SA z/OS 3.5 with APAR level OA51440 and customers using SA MP, and who want to automate SAP components on AIX or Linux using the <i>SAP high availability policy feature</i> .			

IBM documents

Find references to IBM documents related to business continuity and high availability of SAP on IBM z Systems

The latest IBM documentation can be found at:

DB2 IBM Knowledge Center: DB2 for z/OS and IBM DB2 for z/OS

z/OS IBM Knowledge Center: z/OS

IBM Knowledge Center for Tivoli System Automation for z/OS

http://www.ibm.com/support/knowledgecenter/SSWRCJ_3.5.0/com.ibm.safos.doc_3.5/kc_welcome-444.html

Tivoli System Automation for z/OS 3.5.0 documentation (PDF manuals)

http://www.ibm.com/support/knowledgecenter/SSWRCJ_3.5.0/com.ibm.safos.doc_3.5/sa35_product_manuals.html

IBM Knowledge Center for Tivoli System Automation for Multiplatforms

http://www.ibm.com/support/knowledgecenter/en/SSRM2X_4.1.0/com.ibm.samp.doc_4.1/welcome_samp.html

Tivoli System Automation for Multiplatforms 4.1.0 documentation (PDF manuals)

http://www.ibm.com/support/knowledgecenter/en/SSRM2X_4.1.0/com.ibm.samp.doc_4.1/samp_IC_productmanuals.html

Table 44 lists further IBM documents referenced in this publication.

Table 44. Other IBM reference documents

IBM Documents	Order Number
<i>Linux on z Systems Device Drivers, Features, and Commands</i>	SC33-8411
<i>z/OS MVS Setting Up a Sysplex</i>	SA23-1399
<i>Network File System Guide and Reference</i>	SC23-6883
<i>z/OS V2R2 Communications Server: IP Configuration Guide</i>	SC27-3650
<i>z/OS V2R2 Introduction and Release Guide</i>	GA32-0887

Table 45 lists the IBM Redbooks® that you might find useful.

Table 45. IBM Redbooks and Redpapers covering related topics

IBM Redbooks/Redpapers™ (published by the IBM International Technical Support Organization, ITSO)	Order Number
<i>IBM z13 Technical Guide</i>	SG24-8251
<i>IBM System z Strengths and Values</i>	SG24-7333
<i>z/OS V1R13 DFSMS Technical Update</i>	SG24-7961
<i>SAP on DB2 9 for z/OS: Implementing Application Servers on Linux for System z</i>	SG24-6847
<i>IBM Systems for SAP Business Intelligence: 25 Terabyte Scalability Study</i>	REDP-4411
<i>IBM DS8870 Architecture and Implementation (Release 7.5)</i>	SG24-8085
<i>IBM DS8870 Copy Services for IBM z Systems</i>	SG24-6787
<i>IBM GDPS Family - An Introduction to Concepts and Capabilities</i>	SG24-6374
<i>IBM System Storage DS8000: Remote Pair FlashCopy (Preserve Mirror)</i>	REDP-4504
<i>Running SAP Solutions with IBM DB2 10 for z/OS on the zEnterprise System</i>	SG24-7978

SAP documents

Find references to SAP documents related to business continuity and high availability of SAP on IBM z Systems.

You can find the latest SAP documentation at:

SAP on DB2 for z/OS

<https://go.sap.com/community/topic/db2-for-zos.html>

SAP High Availability

<https://wiki.scn.sap.com/wiki/display/SI/SAP+High+Availability>

SAP Installation & Upgrade Documentation

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

SAP Help Portal - The central place for SAP documentation

<http://help.sap.com>

SAP NetWeaver

http://help.sap.com/nw_platform

Table 46. SAP documents

SAP on DB2 for z/OS: Summary of SAP Documentation and Information
Installation publications are available for ABAP-based and Java-based and for mixed ABAP/Java-based (add-in) application servers.
<i>Installation Guide SAP Systems Based on SAP NetWeaver 7.x Application Server on AIX/Linux/Windows: IBM DB2 for z/OS</i>
<i>Planning Guide for SAP on IBM DB2 for z/OS</i>
<i>Database Administration Guide: SAP on IBM DB2 for z/OS</i>
<i>SAP Security Guide for IBM DB2 for z/OS</i>
<i>System Copy Guide SAP NetWeaver 7.5 Including Enhancement Package 1 ABAP</i>
SAP on DB2 for z/OS Community:
https://go.sap.com/community/topic/db2-for-zos.html <ul style="list-style-type: none"> • <i>Casebook 2014 Edition: Tightly Integrated DB2 Backup, Recovery and Cloning for SAP Environments</i> • Unicode conversion • SAP Business Suite on IBM z Systems Reference Architecture • SAP for Banking on IBM z Systems Reference Architecture • SAP for Insurance on IBM z Systems Reference Architecture • Best Practice document: <i>Migrating SAP Systems to DB2 12 for z/OS</i> • <i>SAP HA Installations on z/OS and Windows Application Servers</i> • <i>Rolling Kernel Switch</i>

SAP Notes

This is a list of selected SAP Notes that are referenced in this publication and/or are useful in constructing and maintaining a high availability SAP system on the z Systems platform.

This list serves as a reference to assist you in your availability planning.

SAP Notes can be found on the SAP Support Portal:

<https://support.sap.com/notes>

The complete and most current list of SAP Notes that apply to the SAP on z Systems platform is maintained in *SAP Note 81737*:
<http://service.sap.com/sap/support/notes/81737>

Table 47. Relevant SAP Notes

SAP Note	Title
81737	APAR List
98051	Database Reconnect: Architecture and function
171356	SAP software on Linux: Essential information
353529	Minimizing downtime when switching from and to Daylight Saving Time
538405	Composite SAP Note: SAP Web Dispatcher
768727	Process automatic restart functions in sapstart
809477	startsap/stopsap for SAP WebAs 640, 700, 701, 710, 711, 720
915482	DB2 z/OS: Automating DB failover
951910	NW2004s High Availability Usage Type PI
953653	Rolling kernel switch
1031096	Installing Package SAPHOSTAGENT
1041390	SM21: Central system log via HTTP or HTTPS
1121900	Message server reconnect parameter optimization - AS Java 7.1
1146808	Connection to standalone enqueue server is canceled
1259982	UNIX: Deleting an SAP System Based on NW 7.1 and Higher
1263782	DB2 z/OS: Recommended settings for HiperSockets (zLinux)
1322991	ZSCSinst Installation tool for SAP Central services on z/OS
1391070	Linux UUID solutions
1398993	DB2-z/OS: Running two or more DB2 members on one z/OS LPAR
1414569	DB2 z/OS: Seamless JDBC Failover & Java Monitoring 7.10
1428378	Enqueue server uses a lot of CPU, even though it is idle
1465252	DB2 z/OS: Exploit CLI time out parameter
1496233	ENQU: State transfer between ENSA and ERS
1512505	sapstartsrv: reliable TCP/IP failure recovery
1522391	DB2 z/OS: CCMS: Failover Configuration Tool
1557416	DB2 z/OS: Recommended settings for Gigabit Ethernet
1601565	wdispmon/icmmon does not start
1629558	Linux 3.x kernel
1636252	Installing a 7.20 kernel in SAP Web AS 7.00/7.01/7.10/7.11
1680045	Release Note for Software Provisioning Manager 1.0
1693245	SAP HA Script Connector Library
1704753	Inst.Systems based on NetWeaver 7.1 and higher: UNIX
1744209	SAP-Kernel 720, 721 und 722: Versionen und Kernel-Patch-Levels
1749669	Inst. Systems Based on NW 7.3 and higher
1753638	z/OS: Enqueue Replication into System z Coupling Facility
1777242	DB2 z/OS: CLI client based seamless failover
1787163	Message Server: save logon groups feature

Table 47. Relevant SAP Notes (continued)

SAP Note	Title
1812243	ENQU: Resetting backup file during start
1822055	Enhanced SAPHA library interface
1823660	DB2 z/OS: High Availability with SA z/OS updates and fixes
1855801	DB2 z/OS: vx DB_SET_ISOLATION_LEVEL
1870733	DB2-z/OS: CCMS: Database outside discovery enablement
1881267	DB2-z/OS: SAP Host Agent problem on z/OS
1887279	DB2-z/OS: sapdbctrl: Prerequisites and Configuration
1907566	Obtaining the Latest SAP Host Agent Documentation
1927404	DB2 z/OS: IBM Data Server Driver for CLI/ODBC/JDBC/SQLJ - Special Builds
1969546	Release Roadmap for Kernel 74x and 75x
1973431	sapstart: process restart functions
1984787	SUSE LINUX Enterprise Server 12: Installation notes
2011054	DB2-z/OS: Support status for Seamless CLI Failover Feature
2036171	Missing locks following failover between enqueue server and replicated enqueue server
2047924	DB2-z/OS: CCMS:HAG: CIM-Provider Enablement for z/OS
2073500	FAQ: Splitting off ASCS from PAS
2077934	Rolling kernel switch in HA environments
2081226	Regulations to get the HA Test Tool
2099374	DB2-z/OS: NFSv4 client for Linux on z Systems or Linux on System x is not recommended in a High Availability setup with a z/OS NFS Server movable between LPARs
2101963	DB2 z/OS: Additional authorizations for autobind
2111003	DB2-z/OS: CCMS: db2dsdriver.cfg and dsn_alias
2119669	How to split the ASCS from Primary Application Server (PAS)
2131873	z/OS: Automated Rolling Kernel Switch in HA environment
2177923	Processes started by SAP start service are not auto-restarted when terminated due to error
2187639	Linux 4.x-Kernel
2198998	SAP Kernel 7.22 disp+work (DW.SAR) patch forecast
2239553	DB2-z/OS:CCMS: Installation Parameter Settings for DB2 12
2239846	Rolling kernel switch improvements
2372388	SAP NetWeaver Application Server for ABAP 7.51 Innovation Package: Release Information Note
2380717	SUM in HA environments with *SAPSRV <i>add-on policy</i>

APARs

Find here a reference to a list of APARs for SAP on DB2 for z/OS

For an up-to-date list of all relevant APARs for SAP on DB2 for z/OS, refer to the latest version of *SAP Note 81737*.

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. 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. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user. IBM may have patents or pending patent applications covering subject matter 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
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Deutschland Research & Development GmbH
Department 3282
Schönaicher Strasse 220
D-71032 Böblingen
Federal Republic of Germany
Attention: Information Request

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Any pointers in this publication to Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites. The materials at these Web sites are not part of the licensed materials for SAP on DB2 for z/OS on IBM z Systems. Use of these materials is at your own risk.

Trademarks and service marks

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

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.

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

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

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

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

Glossary

This defines terms used in this publication.

abnormal end of task (abend)

Termination of a task, a job, or a subsystem because of an error condition that cannot be resolved during execution by recovery facilities.

Advanced Interactive Executive (AIX)

The IBM licensed version of the UNIX operating system.

Authorized Program Analysis Report (APAR)

A report of a problem caused by a suspected defect in a current unaltered release of a program. The correction is called an APAR fix. An *Information APAR* resolves an error in IBM documentation or provides customers with information concerning specific problem areas and related.

bind The process by which the output from the DB2 precompiler is converted to a usable control structure called a package or an application plan. During the process, access paths to the data are selected and some authorization checking is performed.

Central Services

See *SAP Central Services (SCS)*

client In commercial, organizational, and technical terms, a self-contained unit in an SAP system with separate master records and its own set of tables.

Cross-System Coupling Facility (XCF)

The hardware element that provides high-speed caching, list processing, and locking functions in a Sysplex.

daemon

A task, process, or thread that intermittently awakens to perform some chores and then goes back to sleep.

data sharing

The ability of two or more DB2 subsystems to directly access and change a single set of data.

data sharing member

A DB2 subsystem assigned by XCF services to a data sharing group.

data sharing group

A collection of one or more DB2 subsystems that directly access and change the same data while maintaining data integrity.

database

A collection of tables, or a collection of table spaces and index spaces.

database host

A machine on which the SAP database is stored and which contains the support necessary to access that database from an instance.

database server

A term that is used for both database host and database service.

database service

A service that stores and retrieves business data in an SAP system.

DB2 Connect

The DB2 product providing client access to a remote database via its IBM Data Server Driver for ODBC and CLI (DB2 CLI driver) and IBM Data Server Driver for JDBC and SQLJ (DB2 JDBC driver) components.

Direct Access Storage Device (DASD)

A device in which the access time is effectively independent of the location of the data.

Distributed Relational Database Architecture™

A connection protocol for distributed relational database processing that is used by the IBM relational database products. DRDA® includes protocols for communication between an application and a remote relational database management system, and for communication between relational database management systems.

Ethernet

A 10- or 100-megabit base band local area network that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by

using collision detection and transmission. Ethernet uses carrier sense multiple access with collision detection (CSMA/CD).

Gigabit Ethernet (GbE)

Gigabit Ethernet is an Ethernet networking standard capable of data transmission rates up to 1000 Mbps. It requires a network interface card (NIC) capable of transmitting data at 1000 Mbps. Gigabit Ethernet can use copper twisted pair wires, coaxial cable, and optical fiber cable as its medium of transmission.

fiber The transmission medium for the serial I/O interface.

File Transfer Protocol (FTP)

The Internet protocol (and program) used to transfer files between hosts. It is an application layer protocol in TCP/IP that uses TELNET and TCP protocols to transfer bulk-data files between machines or hosts.

Extended Binary Coded Decimal Interchange Code (EBCDIC)

A set of 256 characters, each represented by 8 bits.

gateway

Intelligent interface that connects dissimilar networks by converting one protocol to another. The special computers responsible for converting the different protocols, transfer speeds, codes, and so on are also usually considered gateways.

group name

The MVS XCF identifier for a data sharing group.

hexadecimal

Pertaining to a selection, choice, or condition that has 16 possible different values or states.

Pertaining to a fixed-radix numeration system, with radix of 16.

Pertaining to a system of numbers to the base 16; hexadecimal digits range from 0 through 9 and A through F, where A represents 10 and F represents 15.

Hierarchical File System (HFS)

A file system in which information is organized in a tree-like structure of

directories. Each directory can contain files or other directories.

home address

Defines a single virtual IP address that is used by all RS/6000® systems to access z/OS, independent of the number of RS/6000 gateways connected to a given z/OS. This implementation differs from the standard IP model that defines an IP address per physical adapter.

incremental bind

A process by which SQL statements are bound during the execution of an application process, because they could not be bound during the bind process and VALIDATE(RUN) was specified.

Information APAR

An APAR directly related to existing documentation or intended to provide supplementary information.

Initial Program Load (IPL)

The process that loads the system programs from the auxiliary storage, checks the system hardware, and prepares the system for user operations.

instance

An administrative unit that groups together components of an SAP system that provide one or more services. These services are started and stopped at the same time. All components belonging to an instance are specified as parameters in a common instance profile. A central SAP system consists of a single instance that includes all the necessary SAP services.

Internal Resource Lock Manager (IRLM)

A subsystem used by DB2 to control communication and database locking.

Internet

A worldwide network of TCP/IP-based networks.

job Continuous chain of programs, controlled one after the other in time by particular control commands.

Job Control Language (JCL)

A programming language used to code job control statements.

jumbo frame

An Ethernet frame larger than 1518 bytes. Larger frame sizes increase efficiency for data-intensive applications by reducing

frame transmission processing. The maximum frame size is 9000 bytes.

link The transmission medium for the serial I/O interface. A link is a point-to-point pair of conductors (optical fibers) that physically interconnects a control unit and a channel, a channel and a dynamic switch, a control unit and a dynamic switch, or, in some cases, a dynamic switch and another dynamic switch. The two conductors of a link provide a simultaneous two-way communication path. One conductor is for transmitting information and the other is for receiving information. A link is attached to a channel or control unit by means of the link interface of that channel or control unit and to a dynamic switch by means of a dynamic-switch port.

Local Area Network (LAN)

A data network located on the user's premises in which serial transmission is used for direct data communication among data stations.

Logically Partitioned (LPAR) mode

A central processor complex (CPC) power-on reset mode that enables use of the PR/SM™ feature and allows an operator to allocate CPC hardware resources (including central processors, central storage, expanded storage, and channel paths) among logical partitions. Contrast with basic mode.

NetView Management Console (NMC)

A function of the NetView program that provides a graphic, topological presentation of a network that is controlled by the NetView program. It provides the operator different views of a network, multiple levels of graphical detail, and dynamic resource status of the network. This function consists of a series of graphic windows that allows you to manage the network interactively. Formerly known as the NetView Graphic Monitor Facility (NGMF).

Network interface card (NIC)

An expansion board inserted into a computer so the computer can be connected to a network. Most NICs are designed for a particular type of network, protocol, and media, although some can serve multiple networks.

NMC see *NetView Management Console*

Open Shortest Path First (OSPF)

A TCP/IP routing protocol that permits the selection of a specific routing path prior to transmission via IP. It plays an important role in maintaining redundant paths for high availability support.

Path MTU Discovery

A configuration option that requests TCP/IP to dynamically determine the *path MTU*, that is, the minimum MTU for all hops in the path.

proactive redirection

In DB2 data sharing topologies, the need can arise to redirect the work processes of an SAP application server to a different DB2 member of the data sharing group. Optimally, this operation should not be noticed by end users. Therefore, the SAP application server allows the SAP administrator to proactively redirect the work processes to a different DB2 member and thus avoid an error situation. See the *Database Administration Guide: SAP on IBM DB2 for z/OS*.

profile

Summary of system parameters with defined values. The parameters define, for example, the size of buffer areas, the maximum number of system users, and so on. The system parameters can be grouped together in a profile. When activating the system, a certain profile can be called up.

Program Temporary Fix (PTF)

A temporary solution or by-pass of a problem diagnosed by IBM System Support as the result of a defect in a current unaltered release of the program.

Resource Access Control Facility (RACF)

An IBM-licensed product that provides for access control by identifying and verifying users to the system, authorizing access to protected resources, logging detected unauthorized attempts to enter the system, and logging detected accesses to protected resources.

router An intelligent network component that holds information about the configuration of a network and controls data flows accordingly.

SAP SAP AG, a vendor of collaborative

business solutions for a wide variety of industries and markets. The solutions employ an external database management system such as DB2 for z/OS.

SAP Central Services (SCS)

A group of SAP standalone components comprising the

- enqueue server
- message server
- gateway server

Note: SAP also employs the simple abbreviation SCS to designate the Java SCS implementation.

SAP system

An SAP database and a collection of SAP instances (application servers) that provide services to the users. The collection of instances consist of one central instance and, optionally, one or more secondary instances. Each system has a system identifier called SID or SAPSID.

schema

A logical grouping for user-defined functions, distinct types, triggers, and stored procedures. When an object of one of these types is created, it is assigned to one schema, which is determined by the name of the object. For example, the following statement creates a distinct type *T* in schema *C*:

```
CREATE DISTINCT TYPE C.T ...
```

Sysplex Failure Manager (SFM)

The SFM policy can automate the actions to isolate a system that has a status missing condition. This frees up resources, avoiding "sympathy sickness" and data corruption.

SQL Processor Using File Input (SPUFI)

A facility of the TSO attachment subcomponent that enables the DB2I user to execute SQL statements without embedding them in an application program.

Storage Management Subsystem (SMS)

A component of MVS/DFP that is used to automate and centralize the management of storage by providing the storage administrator with control over data class,

storage class, management class, storage group, and automatic class selection routine definitions.

Structured Query Language (SQL)

A standardized language for defining and manipulating data in a relational database.

superuser

A system user who operates without restrictions. A superuser has the special rights and privileges needed to perform administrative tasks.

sysplex failover

Sysplex failover support is the capability of SAP on DB2 to redirect application servers to a standby database server in case the primary database server becomes inaccessible.

System Modification Program Extended (SMP/E)

A licensed program used to install software and software changes on z/OS systems.

Systems Complex (sysplex)

The set of one or more z/OS systems that is given a cross system coupling facility (XCF) name and in which the authorized programs can then use XCF coupling services. A sysplex consists of one or more z/OS systems.

Time-Sharing Option (TSO)

A z/OS option that provides conversational time-sharing from remote terminals.

Transmission Control Protocol/Internet Protocol (TCP/IP)

A software protocol developed for communications between computers.

UNIX System Services

The set of functions provided by the Shell and Utilities, kernel, debugger, file system, C/C++ Run-Time Library, Language Environment®, and other elements of the z/OS operating system that allow users to write and run application programs that conform to UNIX standards.

Virtual IP Address (VIP)

A generic term referring to an internet address on a host that is not associated with a physical adapter.

Virtual Machine (VM)

A functional simulation of a computer and its associated devices. Each virtual machine is controlled by a suitable operating system.

Virtual Storage Access Method (VSAM)

An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry sequence), or by relative-record number.

Term used for storing data on direct-access volumes.

Virtual Telecommunications Access Method (VTAM)

A set of IBM programs that control communication between terminals and application programs.

VSWITCH

z/VM Virtual Switch, a z/VM networking function, introduced with z/VM 4.4, that provides IEEE 802.1Q VLAN support for z/VM guests. It is designed to improve the interaction between guests running under z/VM and the physical network connected to the z Systems processor.

Workload Manager (WLM)

The workload management services enable z/OS to cooperate with subsystem work managers to achieve installation-defined goals for work to distribute work across a sysplex, to manage servers and to provide meaningful feedback on how well workload management has achieved those goals. They also allow programs to create an interface to define a service definition. To change from resource-based performance management to goal-oriented workload management, many transaction managers, data managers, and performance monitors and reporters need to take advantage of the services z/OS workload management provides.

Work Process (WP)

A job in the SAP system that actually does the work. Each work process is assigned a primary role by the dispatcher,

which controls, to a certain degree, what type of work is to be performed by that work process. The number of work processes and the types that can exist for an instance are controlled by the instance profile and within the SAP system by the Central Computer Management System.

Zebra An open-source (GNU) routing package that manages TCP/IP based routing protocols. In the high availability solution for SAP, it enables the functions of the Open Shortest Path First (OSPF) routing protocol on Linux on z Systems.

z Systems

A range of IBM mainframe processors representing the successors to the S/390®.

Index

Special characters

- *DB2 add-on policy 95
- *SAP sapstartsrv add-on policy
 - defining in Tivoli System Automation for z/OS 149
- *SAPSRV add-on policy 149, 153
 - shipped with SA 3.5 APAR OA48922 145
 - summary of changes xi, xii
- configuration file 231
- wicked framework 300

A

- ABAP application server
 - check_as script 137
 - configuring for SA z/OS 137
 - shell scripts 137
 - start_as script 137
 - stop_as script 137
- ABAP central services 164
 - and enqueue replication server 159
- ABAP enqueue server
 - dependencies to enqueue replication server 162
- ABAP instances
 - mass move 261
- ABAP SAP Central Services
 - configuring and starting 136
- ABAP SCS
 - ASCS 120
 - mandatory manual modifications after Java SCS installation 123
 - setting up 116
- ABAP SCS with enqueue replication
 - verifying 122
- ABAP type SAP instances
 - DB2 connection failover 16
- ABAP-only application servers 174
- active automation
 - running the RKS 274
- active connections
 - checking 236
- adapter teaming 54
- administrator home directory 99
- AIX
 - application server timeout behavior 59
 - Source VIPA 46, 48
- AIX application server 44
- AIX OSPF 299
- AIX utilities 285
- APAR OA48922
 - *SAPSRV add-on policy 145
- APARs
 - list of 321
- application design
 - SAP high availability 83
- application server
 - check_as script 288
- application server (*continued*)
 - dual-stack (ABAP plus Java) 174
 - on AIX
 - timeout behavior 59
 - on Linux on z Systems
 - timeout behavior 62
 - on Windows
 - timeout behavior 63
 - rfcping_ibm 137
 - start_as script 287
 - start_cs script 288
 - stop_as script 287
 - Windows 129
- application server (SAP)
 - as SAP resource 171
- application server group
 - as SAP resource 174
- application server instances
 - additional 128
 - installing additional ones 129
 - Java-only 129
 - primary 128
- application servers
 - ABAP-only 174
 - controlled by SA MP 249
 - controlled remotely 249
- applications
 - checking for problems 234
 - health check 5
- architecture
 - database server 81
 - file system 78
 - network 74
 - of high availability solution 69
- architecture components
 - for SAP high availability 69
- AS to DB and SCS connections 59
- ASCS
 - ABAP SCS 120, 122
 - and enqueue replication server 159
 - configuring and starting 136
- autofs script 107
- automated verification
 - NFS HA setup 308
- Automatic I/O Interface Reset Facility 109
- automatic restart
 - sapstartsrv 206
- automation
 - objectives for SAP 5
- automation names 151
- automation scripts 285
- automount samples
 - NFS client 301
- autonomic computing
 - self-managing systems 1, 11
- availability
 - with data-sharing configuration 81
 - with non-data-sharing configuration 81

- availability features
 - Parallel Sysplex 12
 - z/OS 11

B

- backup and recovery
 - with data sharing 32
- BACKUP SYSTEM utility
 - DB2 for z/OS 32
- benefits
 - for SAP availability 11
- bibliography 317
 - IBM documents 318
 - SAP documents 319
- Business Continuity for SAP on IBM z Systems
 - edition history 317
 - supported component versions overview and history 317
- business continuity solution
 - sample network setup 293
- business continuity solution for SAP on IBM z Systems 1

C

- C-shell 136
- Central Electronic Complex (CEC) 13
- central instance
 - DB2 connection failover 83
 - double network 83
 - replaced by SAP Central Services 70
 - with data sharing 83
 - without data sharing 83
- Central Services
 - installing with ZSCSinst 118
- change management 249
 - DB2 249
 - DB2 CLI driver 265
 - DB2 JDBC driver 265
 - for SAP on z/OS UNIX 264
 - on SAP central services hosts 267
 - operational points 248
 - SAP application servers 265
 - SAP kernel 249
 - z/OS 249
- check_as script 138
- checkwd 290
- CIM provider component
 - of the SAP host agent 105
- classes in SA z/OS 153
- cleanipc 281
- cleanrepstzOSCf 281
- CLI driver 16
- CLI failover 16
- client
 - connection timeout 62
 - AIX 59

- client (*continued*)
 - idle timeout
 - AIX 59
 - Linux on z Systems 62
 - Windows 63
 - transmission timeout 62
 - TCP/IP on AIX 59
- client/server connection over TCP/IP
 - timeout behavior 59
- commands for testing and debugging 309
- conditions in relationships 148, 172
- config.xml 249
- configuration file
 - db2dsdriver.cfg 130
 - update 130
- configuring the NFS client for NFSv4 305
- configuring the NFS server
 - z/OS 305
- configuring the NFS server for NFSv4 305
- configuring the NFSv4 client on AIX 306
- connect.ini 249
- connection failover 13
- connection status
 - DB2 for z/OS 239
 - SAP 239
- connection timeout
 - Linux on z Systems client 62
- connections
 - checking 236
- considerations for network setup
 - AIX application server 44
 - Linux on z Systems application server 42
 - Linux on z Systems guests under z/VM 42
 - z/OS communication software 41
- continuous availability 3
- continuous operation 3
- conventions 8
- Coupling Facilities (CF) 13
- coupling facility
 - enqueue replication 277
- CPUPLUGD utility 187
- Cross Stack EtherChannel 54

D

- data integrity verification scenario 307
- data sets
 - PROFILE.TCPIP 65
- data sharing 28
- database server
 - architecture for high availability 81
 - idle timeout 65
 - transmission timeout 65
- Database Shared Library (DBSL) 16
- DB connection failover 39
- DB2
 - connection failover architecture 13
- DB2 Analytics Accelerator 28
- DB2 CLI driver 15, 64
 - update 265
- DB2 client connections
 - timeout behavior 64

- DB2 Connect
 - update 265
- DB2 connection failover 15
 - central instance 83
 - for ABAP instances 16
- DB2 data sharing 28
 - architecture 13
 - availability considerations 81
 - backup and recovery architecture 32
 - central instance 83
 - central instance without 83
 - considerations for disaster recovery 32
 - on Parallel Sysplex 13
- DB2 database server group (SAPHA1_DBX)
 - as SAP resource 169
- DB2 exception events
 - deadlocks 65
- DB2 for z/OS 185
 - BACKUP SYSTEM utility 32
 - check if running 239
 - checking connection status 239
 - data sharing 81
 - net work for SAP high availability 293
 - non-data-sharing 81
 - non-disruptive software changes 12
 - planning information 95
 - updating 249
- DB2 JDBC driver 15, 64
 - update 265
- DB2 policy 169
- DB2 server driven failover
 - with native CLI driver 16
- db2_sazoscmd.rex 290
- db2dsdriver.cfg
 - update 130
- DBACOCKPIT 105
- DBSL
 - Database Shared Library 16
- DDF
 - keep-alive interval times 65
- DDS 104
 - binding with DVIPA 105
- deadlock detection interval 65
- debugging commands 309
- degrees of availability 3
- deleting an SMDA instance 135
- dependencies
 - dependencies to ABAP enqueue server 162
 - enqueue replication server 162
 - enqueue server 162
- disaster recovery
 - data sharing considerations for 32
 - GDPS infrastructure 32
 - tracker site 32
- DISPINFO 231
- disruptive procedure
 - updating the SAP kernel 272
- Distributed Data Server 104
- dual-stack (ABAP + Java) installation
 - process 114
- dual-stack (ABAP plus Java) application server 174

- DVIPA
 - binding with DDS 105
 - setting up 105
- dynamic VIPA 46, 236

E

- edition history 317
 - Business Continuity for SAP on IBM z Systems 317
- end point problem
 - of a TCP/IP connection 46
- EnqCF replication 277
 - adapting the SAPSRV add-on policy 278
 - failover and recovery 73
 - trouble shooting 282
- EnqCF replication policy changes 149
- EnqCF replication tool support 281
- enquelog trace file 167, 174
- enqueue replication 167
 - into a z Systems coupling facility 277
- Java SCS 124
 - monitoring health state under SA z/OS 174
 - verifying ABAP SCS 122
- enqueue replication server 174
 - and ABAP central services 159
 - and Java central services 164
 - failure of sapstart process 216
 - failure scenario 83
 - installing as an ERS instance 118
 - installing on z/OS 118
 - installing with ZSCSinst 118
- enqueue server 174
 - dependencies to enqueue replication server 162
 - failure 216
- entry names 151
- environment
 - AIX 106
 - AIX modifying 106
 - Linux on z Systems 106
 - Linux on z Systems modifying 107
- ERS 174
- ERS instances
 - mandatory manual modifications after Java SCS installation 123
- EtherChannel 54
- even number of machines
 - using quorum nodes 189

F

- failover
 - CLI failover 16
 - DB2 connection 15
 - DB2 server driven 16
 - of NFS server 78
 - of SAP Central Services 70
 - SAP failover 16
- failover of SAP Central Services
 - using EnqCF replication 73
 - using replication into a z Systems coupling facility 73

- failover scenarios
 - Linux on z Systems 241
 - SA z/OS policy 193
- failover verification scenario 307
- failure
 - of an LPAR 216
 - of enqueue server 216
 - of message server 206
 - of NFS server 216
 - of TCP/IP stack 216
 - SAP start service 216
- failure of sapstart process
 - enqueue replication server 216
 - SAP Central Services 216
- failure scenarios
 - impact on SAP system 83
- Fast Lookup Cache (FLC) 98
- fault tolerant network 74
- file samples
 - z/OS 309
 - z/OS UNIX 309
- file system
 - architecture 78
 - moving to specific LPAR 101
 - NFS 238
 - planning information 96
 - pseudo-root (NFSv4) 304
 - setup 301
 - shared HFS 238
 - sysplex-aware 98
- file systems
 - location 96
- FLC 98
- full-blown high availability implementation 21

G

- gateway service - optional component 164
- GDPS
 - infrastructure for disaster recovery 32
- general installation sequence 114
- generate a policy
 - SAP high availability policy feature 186
 - using the sample SAP HA policy 187
- GetFileSystemOwnership 101, 290
- GetWebPage 5
- global directories 96
- global transport directory 99
- group structure 152
- groups
 - SAP system-dependent 158
 - SAP system-independent 155
- groups and applications 180

H

- HA Option 1 24, 25
- HA Option 2 24, 27
 - implementing 185
- hardware considerations for network setup 41
- HATT 194

- health check
 - applications 5
 - SAP Web Dispatcher 132
- health state of SAP enqueue replication monitoring under System Automation 167
- high availability 3
 - definitions 3
 - objectives for SAP 5
 - recommended setup
 - OSPF 48
 - recovery attributes 48
 - recommended setup for client/server connections 48
 - SAP sysplex failover 45
 - High Availability (HA) Test Tool 194
 - high availability library interface for SA z/OS 168
 - high availability scripts
 - for IBM Tivoli System Automation for z/OS 285
 - high availability solution
 - sample configuration 89
 - high availability solution for SAP
 - architecture 69
 - automation 6
 - overview 6
 - planning 21
 - planning and preparing 89
 - software prerequisites 90
 - high availability solution for SAP on IBM z Systems
 - overview 1
 - hints and tips
 - NFSv4 migration 302
 - hints on Linux SLES 11 SP2 313
 - hints on Linux SLES 12
 - together with SA MP 315
 - hints on Linux SLES 12 SP1 and SP2 315
 - HiperSockets 28, 41
 - host agent file system 99

I

 - IBM Data Server Driver for JDBC and SQLJ 15
 - IBM Data Server Driver for ODBC and CLI 15
 - IBM DB2 for z/OS 185
 - IBM Ported Tools 140
 - IBM Tivoli System Automation for z/OS
 - high availability scripts 285
 - IBM z Systems 277
 - EnqCF replication 277
 - enqueue replication into a z Systems coupling facility 277
 - idle timeout
 - database server 65
 - Linux on z Systems client 62
 - IEDN 39, 41, 48
 - implementation
 - SA MP policy 186
 - implementation on z/OS
 - test scenarios 193
 - Implementing Option 2 185
 - in place 278
 - infrastructure group 155

- infrastructure group (*continued*)
 - stopping and starting 209
- ING_sap.tar 285
- INGEVIPA 143
- INGINFO 231
- INGNFSGS 290
- installation sequence 114
- installing additional SAP application server instances
 - Java-only 129
- installing SAP primary application server 129
- installing the enqueue replication server 118
- Internal Resource Lock Manager (IRLM) 65
- iptables
 - as alternative to SourceVIPA 48

J

- Java application server instance
 - start a remote instance 141
- Java central services 116
 - and enqueue replication server 164
- Java Central Services
 - configuring and starting 137
- Java instances 18, 19
- Java SCS 116
 - configuring and starting 137
 - enqueue replication 124
 - mandatory manual modifications after Java SCS installation 123
 - setting up 116
- Java-only application servers
 - resources 174

K

- keep-alive
 - DDF 65
 - probes 59, 63, 65

L

- Large Fast Lookup Cache (FLC)
 - Buffer 98
- Large FLC Buffer 98
- light restart 95
- Link State Advertisements 45
- Linux
 - SL12 together with SA MP 313
 - SLES 11 SP2 313
 - SLES 11 to SLES 12 migration 313
 - SLES 12 315
 - SLES 12 SP1 315
 - SLES 12 SP2 315
- Linux on System x 129
- Linux on z Systems
 - application server timeout behavior 62
 - failover scenarios 241
 - multiple guests under z/VM 41
 - network settings 298
 - Quagga setup 298
 - verification 241

- Linux on z Systems *(continued)*
 - Zebra setup 298
- Linux on z Systems application server 42
- Linux on z Systemsguests under z/VM 42
- Linux SLES 11 SP2 313
- Linux SLES 12
 - SP1 313
 - SP2 313
- Linux utilities 285
- load balancing
 - OSPF 45
- lock manager
 - NFSv4 304
- logon profile
 - configuring SAP on SA z/OS 136
- LPAR
 - failure 216
 - shutdown and restart 206
- LPAR-to-LPAR communication 41

M

- maintenance requiring reboot
 - of an SAP application server 269
- Management Console 24
- manual modifications
 - after ASCS installation 120
 - mandatory after Java SCS installation 123
- mass move
 - ABAP instances 261
- Message Processing Facility (z/OS) 231
- message server
 - failure 206
- migrating to SLES 12 315
- Migrating to SLES 12 SP2 315
- migration
 - NFSv4 302
- migration considerations 293
- migration from SLES 11 to SLES 12 313
- monitoring health state
 - of SAP enqueue replication 174
- monitoring health state of SAP enqueue replication
 - under System Automation 167
- multipath routing 39
- multiple SAP ABAP application servers
 - switching database connections 261

N

- naming conventions 91
 - Tivoli System Automation for Multiplatforms 95
 - Tivoli System Automation for z/OS 92, 94
- naming conventions for SAP
 - resources 151
- native CLI driver 16
- NetView
 - netlog 231
 - planning information 146
- NetView Management Console 158, 231
- network 39

- network *(continued)*
 - alternative setup 54
 - architecture considerations 74
 - central instance 83
 - fault tolerant 74
 - hardware 294
 - problem determination 236
 - setup 293
 - setup recommendations 41
- network attributes
 - AIX 59
 - Linux on z Systems 62
 - tcp_keepalive_intvl 62
 - tcp_keepalive_probes 62
 - tcp_keepalive_time 62
 - tcp_retries2 62
 - Linux on z Systems application server 62
 - tcp_syn_retries 62
- network failures
 - impact levels 39
- Network File System
 - NFS 303
- Network File System server
 - NFS server 6
- network performance
 - RSCT 188
- network settings
 - z/OS 294
- network setup
 - for a business continuity solution 293
 - Linux on z Systems 298
 - z/OS settings 294
- NFS
 - checking status 238
 - Network File System 303
- NFS client 103
 - automount samples 301
 - configuring for NFSv4 305
 - root access 101
 - verifying for NFSv4 305
- NFS client adaption 313
- NFS failover 83
- NFS HA setup
 - automated verification 308
- NFS high availability policy 28, 104
- NFS server 155
 - configuring for NFSv4 305
 - failover 78
 - failure 216
 - Network File System server 6
 - nlm site attribute 101
 - non z/OS 104
 - on z/OS 101
 - remount attribute 101
 - restimeout site attribute 101
 - security model 101
 - verifying for NFSv4 305
- NFSv3
 - NFS version 3 78
- NFSv3 mounts
 - Linux sample 301
- NFSv4
 - configuring 305
 - features 303
 - lock manager 304

- NFSv4 *(continued)*
 - migration 303
 - NFS version 4 78, 305
 - pseudo file system 304
 - symbolic links on z/OS 304
- NFSv4 client 313
- NFSv4 features 303
- NFSv4 migration 302
- NFSv4 mounts
 - AIX sample 302
- NFSv4 related information 303
- NIC
 - failure recovery 46
- NIC failure recovery
 - subnet configuration 48
 - VIPA 48
- nlm site attribute 101
- NMC 158, 231
- non-data-sharing
 - availability considerations 81
- non-disruptive software changes
 - DB2 for z/OS 12

O

- odd number of machines
 - using quorum nodes 190
- OMPROUTE
 - network configurations with and without OMPROUTE 148
- OMPROUTE policy changes 148
- Open Shortest Path First (OSPF)
 - as recovery mechanism 39, 45
 - configuration aspects 48
 - dead router interval 48
 - gated daemon sample definition 299
 - implementation 45
 - load balancing 45
 - tables 236
- OpenSSH 140
- operating an SAP system 243
- Option 1 24
- Option 2 24
 - implementing 185
- optional component of ABAP central services 164
- OSPF 299
- ospfd process 107
- outages
 - planned 4
 - unplanned 4

P

- PAM
 - SAP Product Availability Matrix 185
- Parallel Sysplex
 - architecture 13
 - availability features 12
 - DB2 data sharing 13
- parameters
 - AIX
 - rto_high 59
 - rto_length 59
 - rto_limit 59
 - rto_low 59

- parameters (*continued*)
 - AIX (*continued*)
 - tcp_keepidle 59
 - tcp_keepinit 59
 - tcp_keepintvl 59
 - SAP profile parameters
 - rdisp/max_wprun_time 67
 - TCP/IP on Windows 63
 - Windows
 - KeepAliveInterval 63
 - KeepAliveTime 63
 - TcpMaxConnect
 - Retransmissions 63
 - TcpMaxDataRetransmissions 63
- PAS
 - installation 114, 129
 - primary application server 114, 129
- path MTU discovery
 - multipath routing 39
- Path MTU discovery 44
- planned outage 4
- planned outages 206
- planning for SAP high availability 69
- platform specific recommendations
 - SA MP 187
- policy changes
 - EnqCF replication 149
 - OMPROUTE 148
 - TCPIP 148
- policy generation
 - SAP high availability policy
 - feature 186
- policy-based self-healing 5
- Ported Tools for z/OS 140
- PORTMAP
 - migration to RPCBIND 312
- post installation steps 130
- Poststart command 101
- preparation stages of high availability 21
- preparing for the test 201
- prerequisites for SAP high availability 69
- primary application server
 - installation 114, 129
 - PAS 114, 129
- primary application server instances 128
- problem determination
 - Linux/AIX 241
 - SA z/OS 231
 - SAP components 231
 - Tivoli System Automation for z/OS 231
 - z/OS 193
- problem handling with EnqCF replication
 - runtime 282
 - startup 282
- profile parameters for SAP 126
- pseudo-root file system 304
- purpose of a test 197

Q

- Quagga setup 298
- quorum nodes
 - even number of machines 189
 - odd number of machines 190

- quorum, System Automation for Multiplatfroms 188

R

- readme file 285, 290
- recovery
 - of SAP Central Services 70
 - remote using archive logs 32
- recovery mechanisms
 - DB connection failover 39
 - dynamic routing (OSPF) 39
 - on Windows 54
 - OSPF 45
 - SAP sysplex failover 45
 - Virtual IP Addresses (VIPAs) 39, 46
- recovery of SAP Central Services
 - using EnqCF replication 73
 - using replication into a z Systems coupling facility 73
- recovery site
 - configuring 32
- references 317
- registry values
 - Windows 63
- relationships 180
 - with or without a condition 148, 172
- remote application server policy
 - creating 172
 - overview of the relationships 172
 - scripts 174
- remote application servers
 - overview of the relationships 172
- remote control
 - of SAP application servers 249
- remote execution 140
 - of scripts 137
- remote file system access 97
- remote Java application 141
- remote site recovery 32
- remount attribute 101
- replication
 - enqueue 277
 - TCPIP 277
- replication status
 - checking 248
- resource naming conventions 151
- resource status
 - verifying 197
 - verifying the resource status 197
- resource timeout 65
- resources for Java-only application servers 174
- restart in place
 - on the same LPAR 280
- Restart_Program_XX 132
- restarting SAP application server
 - SAP tools 268
- restimeout site attribute 101
- rfcping utility 5
- RKS
 - updating SAP kernel 272
- RMF DDS 104
- RMF Distributed Data Server 104
- rolling kernel switch
 - running with active automation 274

- rolling switch
 - of SAP kernel 249
- route permission file 143
- routing tables 236
- RPCBIND
 - migration from PORTMAP 312
- RPCBIND recommendations 148
- RSCT network performance 188
- running the test 205
- runtime problem handling
 - with EnqCF replication 282

S

- SA 3.5 APAR OA48922
- *SAPSRV add-on policy 145
- SA MP 185
 - naming conventions 95
 - platform specific recommendations 187
- SA MP policy
 - implementation 186
- SA z/OS
 - checking for problems with UNIX applications 234
- sample environment 304
- sample high availability solution
 - configuration for SAP 89
- sample network setup 293
- sample SAP HA policy 187
- sample scripts 285, 290
- sanchkv1.txt 290
- SAP
 - administrator home directory 99
 - checking connection status 239
 - checking database connections 239
 - command summary for Tivoli System Automation 143
 - configuring for Tivoli System Automation for z/OS 135
 - customizing for high availability 113
 - directory definitions 99
 - global transport directory 99
 - high availability and automation objectives 5
 - host agent file system 99
 - license considerations 110
 - logon groups 111
 - starting with SA MP 243
 - system-wide directories 99
 - upgrade procedure 249
- SAP application server
 - maintenance requiring reboot 269
- SAP application server group 174
- SAP application server instances
 - installing additional ones 129
 - Java-only 129
- SAP application servers
 - change management 265
 - controlled by SA MP 249
 - controlled remotely 249
 - stopping 243
- SAP availability
 - z Systems 11
- SAP central services
 - ABAP stop 243
 - DB2 stop 243

- SAP central services (*continued*)
 - Java stop 243
- SAP Central Services 1
 - failover 70
 - failure of sapstart process 216
 - failure scenario 83
 - installing on z/OS 118
 - policy migration to *SAPSRV add-on policy 311
 - recovery 70
 - replacement for central instance 70
 - seamless integration 24
- SAP central services hosts
 - change management 267
- SAP change management
 - z/OS UNIX 264
- SAP Community Network
 - SCN xiii, 96
- SAP enqueue replication
 - EnqCF replication 277
 - method available for SAP on z Systems 277
 - monitoring health state under SA z/OS 174
 - monitoring the health state 167
 - standard method 277
- SAP failover 16
- SAP global directories 96
- SAP HA Interface
 - installation 168
 - overview 168
 - setup 168
- SAP HA Interface for SA z/OS 168
- SAP HA library interface for System Automation
 - installation 168
 - overview 168
- SAP HA wizard 149
- SAP high availability
 - application design 83
 - architecture components 69
 - planning 69
 - prerequisites 69
- SAP high availability policy feature 186
 - to generate a policy 186
- SAP High Availability Test Tool 194
- SAP High Availability wizard 149
- SAP host agent
 - CIM provider component 105
 - starting/stopping via System Automation 142
- SAP Host Agent group 155
- SAP infrastructure group 155
 - stopping and starting 209
- SAP infrastructure tools
 - Management Console 24
 - Software Update Manager (SUM) 24
- SAP installation
 - planning information 109
- SAP instances
 - ABAP instances 16
 - DB2 connection failover 16
 - Java instances 18, 19
- SAP integrity verification scenario 307
- SAP kernel
 - rolling switch 249
 - updating 271
- SAP lifecycle tools 186
- SAP Management Console 24, 168
- SAP MC 168
- SAP Notes
 - list of 319
- SAP on DB2 for z/OS 293
- SAP primary application server
 - installation 129
- SAP Product Availability Matrix PAM 185
- SAP profile parameters 126
- SAP resources
 - classes 153
 - DB2 database server group (SAPHA1_DBX) 169
 - overview 150
- SAP Software Update Manager SUM 270
- SAP start service
 - failure 216
- SAP system
 - failure scenarios 83
 - stopping with SA z/OS 243
- SAP system identifier
 - SAPSID 8
 - SID 8
- SAP system name
 - SAPSID 8
 - SID 8
- SAP system-dependent groups
 - purpose, contents, naming conventions 158
- SAP system-independent groups 155
- SAP tools
 - restarting SAP application server 268
- SAP transactions
 - maximum time 67
- SAP Web Dispatcher 131
 - health check 132
 - high availability 132
 - installing 132
 - SA z/OS policy 132
- SAP Web Dispatcher group 155
- sapcimb 142
- SAPHOST_AGENT 155
- SAPHOST_AGT 155
- saphostexec 142, 150
- saposcol 142
- SAPoscol group 155
- SAPRASTP REXX script 141, 174, 206, 243, 249
- saproot 142
- SAProuter
 - route permission file 143
- SAProuter group 155
- SAPSRV add-on policy
 - adapting for EnqCF replication 278
- sapstart process
 - failure for enqueue replication server 216
 - failure for SAP Central Services 216
- sapstartsrv 132, 136
 - automatic restart 206
- SCN
 - SAP Community Network xiii, 96
- script availability 285
- scripts
 - automation scripts 285
 - high availability 285
 - sample scripts 285
- SCS 1
 - configuring and starting 137
 - Java central services 116
 - Java SCS 116
 - policy migration to *SAPSRV add-on policy 311
- security(exports) 101
- self-healing 5
- self-managing systems 1
- Server Time Protocol (STP) 13
- Service Management Unite 146
- SFM policy 109
- Shared HFS
 - checking status 238
- shell scripts 138
- shutdown problem 315
- SID 8
- single SAP ABAP application servers
 - switching database connections 251
- single-stack (ABAP or Java) installation
 - process 114
- SLES 11 SP2 313
- SLES 11 to SLES 12 migration 313
- SLES 12
 - migrating 315
 - SP1 313
 - SP2 313
- SLES 12 SP1 315
- SLES 12 SP2 315
- SMDA 113
 - deletion 135
 - installation 134
 - SAPSIDSMD group 155
 - SAPSIDSMSRV group 155
 - software prerequisites 90
 - Solution Manager Diagnostics Agent 6, 133, 134, 135, 155
- software prerequisites 90
- Software Update Manager SUM 270
- Software Update Manager (SUM) 16, 24, 186
- Solution Manager Diagnostics Agent 113
 - deletion 135
 - installation 134
 - SAPSIDSMD group 155
 - SAPSIDSMSRV group 155
 - SMDA 6, 133, 134, 135, 155
 - software prerequisites 90
- Source VIPA
 - iptables as alternative 48
 - on AIX 46, 48
 - on remote application servers 48
- SSD
 - system status detection 109
- SSH setup
 - start_as 285
 - stop_as 285
- ssh-rand-helper 140
- stand-alone enqueue server 1
- standalone enqueue server 70
- START DB2 command
 - LIGHT option 95

- start_as
 - SSH setup 285
 - start_as script 138
 - start_sapsrv 289
 - starting
 - SAP system 243
 - starting an SAP system
 - with SA z/OS 245, 248
 - with System Automation for Multiplatforms 248
 - starting an SAP system with SA z/OS 243
 - startup problem handling
 - with EnqCF replication 282
 - static VIPA 46
 - static VIPA definitions
 - for SUSE 300
 - Stop SAP central services
 - ABAP 243
 - DB2 243
 - Java 243
 - stop_as
 - SSH setup 285
 - stop_as script 138
 - stopping
 - SAP system 243
 - stopping an SAP system
 - with SA z/OS 243, 246
 - with System Automation for Multiplatforms 246
 - stopping SAP application servers 243
 - subsystem names 151
 - SUM
 - Software Update Manager 16, 186, 270
 - summary of changes
 - *SAPSRV add-on policy xi, xii
 - supported component versions
 - overview and history 317
 - SUSE
 - static VIPA definitions 300
 - switching database connections
 - multiple SAP ABAP application servers 261
 - single SAP ABAP application servers 251
 - SWPM 16
 - syslogd 147
 - sysplex failover
 - as recovery mechanism 45
 - sysplex failure management
 - policy 109
 - SFM 109
 - sysplex-aware file system 98
 - System Automation
 - self-healing technologies for autonomic computing 5
 - starting/stopping the SAP host agent 142
 - System Automation for Multiplatforms 185
 - CPUPLUGD utility 187
 - customizing 185
 - Linux on z Systems environment 187
 - platform specific recommendations 187
 - System Managed Duplexing 278
 - system status detection
 - SSD 109
 - System System Automation for Multiplatforms 188
 - system-wide directories 99
 - systemd adaption 314
- ## T
- taproot user 105
 - TCP/IP
 - failure 216
 - TCP/IP connection
 - end point problem 46
 - TCPIP policy changes 148
 - TCPIP-based replication 277
 - technologies on IBM z Systems 28
 - terminology 8
 - test environment 197
 - test methodology
 - high availability tests for SAP on z Systems 197
 - test preparing 201
 - test purpose 197
 - test running 205
 - test scenarios
 - for high availability of SAP on z Systems on z/OS 206, 216
 - implementation on z/OS 193
 - planned outages 206
 - unplanned outages 216
 - test verification 205
 - testing commands 309
 - testNFS 290
 - timeout behavior
 - client/server connection over TCP/IP 59
 - Linux on z Systems application server 62
 - of AIX applicationserver 59
 - of database server 65
 - of DB2 client connections 64
 - of Windows application server 63
 - Tivoli System Automation
 - planning information 109
 - self-healing technologies for autonomic computing 5
 - Tivoli System Automation for Multiplatforms
 - customizing 185
 - documentation 318
 - naming conventions 95
 - Tivoli System Automation for z/OS
 - adapting the SA z/OS 3.4 *SAPSRV add-on policy 149
 - check_as script 288
 - configuring SAP for 135
 - customizing 145
 - documentation 318
 - high availability benefits 6
 - naming conventions 92, 94
 - planning information 146
 - preparing for high availability 146
 - problem determination 231
 - start_as script 287
 - start_cs script 288
 - stop_as script 287
- ## U
- UNIX applications
 - checking for problems 234
 - UNIX messages
 - sending to NetView 231
 - sending to syslog 147, 231
 - UNIX System Services
 - setup 294
 - unplanned outage 4
 - unplanned outages
 - for high availability of SAP on z Systems on z/OS 216
 - test scenarios 216
 - update
 - DB2 CLI driver 265
 - DB2 JDBC driver 265
 - update db2dsdriver.cfg 130
 - updating SAP kernel
 - RKS 272
 - updating the SAP kernel 271
 - using a disruptive procedure 272
 - upgrade
 - procedure for SAP 249
 - USS PARMLIB member BPXPRMxx 101
 - USS zFS, file system type 97
 - utilities
 - AIX 285
 - Linux 285
 - z/OS UNIX 285
- ## V
- verification
 - Linux on z Systems 241
 - Linux/AIX 241
 - SA z/OS policy 193
 - z/OS 193
 - verification scenario
 - data integrity 307
 - failover 307
 - SAP integrity 307
 - verifications after test 205
 - verify zFS settings 97
 - verifying ABAP SCS
 - with enqueue replication 122
 - verifying the NFS client for NFSv4 305
 - verifying the NFS server for NFSv4 305

- vipa
 - source vipa on aix 48
- VIPA
 - AIX considerations 48
 - as recovery mechanism 39, 46
 - dynamic 46, 58
 - Linux considerations 48
 - Source VIPA on remote application servers 48
 - static 46, 58
 - z/OS 58, 294
- VIPA definitions
 - for SUSE 300
- Virtual IP Address
 - See VIPA 46
- virtual operating systems 188
- Virtual Switch (VSWITCH) 41

W

- Web Dispatcher 132
- Web Dispatcher group 155
- Windows
 - registry values for timeout 63
- Windows application server 129
- Windows client
 - connection timeout 63
 - transmission timeout 63
- wizard
 - SAP HA wizard 149
 - SAP High Availability wizard 149

Z

- z Systems
 - Parallel Sysplex features and benefits 12
 - SAP availability benefits 11
- z/FS file system
 - performance improvements 98
- z/OS
 - availability features 11
 - configuring the NFS server 305
 - high availability test scenarios for SAP on z Systems 193
 - Message Processing Facility 231
 - networking software 41
 - NFS server on 101
 - non-disruptive software changes 12
 - symbolic links 304
 - syslog 231
 - updating 249
 - VIPA 58, 294
- z/OS and SFM policy 109
- z/OS communication software 41
- z/OS file samples 309
- z/OS network settings 294
- z/OS Solution Manager Diagnostics Agent
 - installation 134
 - SMDA 134
- z/OS UNIX 114, 264
 - SAP change management 264
- z/OS UNIX file samples 309
- z/OS UNIX System Services 114, 294
 - z/OS UNIX System Services syslog daemon 147
 - z/OS UNIX utilities 285
- z/VM
 - multiple Linux on z Systems guests 41
 - Virtual Switch (VSWITCH) 41
- zBX
 - IEDN 39, 41, 48
- zebra process 107
- Zebra setup 298
- zEnterprise
 - IEDN 39, 41, 48
- zFS setting
 - verification 97
- zFS, file system type (z/OS UNIX) 97
- zSAP_BusinessContinuity.zip 187
- zSAP_BusinessContinuitynn.zip 285
- ZSCSinst 114, 118
 - installing SAP Web Dispatcher 132
 - invocation 132

Readers' Comments — We'd Like to Hear from You

SAP on z Systems
Business Continuity for SAP on IBM z Systems
Edition 12/2016

Publication No. SC33-8206-09

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send your comments via email to: s390id@de.ibm.com

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

Email address



Fold and Tape

Please do not staple

Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Deutschland Research & Development GmbH
Information Development
Department 3282
Schoenaicher Strasse 220
71032 Boeblingen
Germany

Fold and Tape

Please do not staple

Fold and Tape



SC33-8206-09

